

**MORE PRACTICAL SKETCHING ALGORITHMS FOR LOW-RANK  
MATRIX APPROXIMATION**

**JOEL A. TROPP, ALP YURTSEVER,  
MADELEINE UDELL, AND VOLKAN CEVHER**

Technical Report No. 2018-01  
October 2018

APPLIED & COMPUTATIONAL MATHEMATICS  
CALIFORNIA INSTITUTE OF TECHNOLOGY  
mail code 9-94 · pasadena, ca 91125

# MORE PRACTICAL SKETCHING ALGORITHMS FOR LOW-RANK MATRIX APPROXIMATION\*

JOEL A. TROPP<sup>†</sup>, ALP YURTSEVER<sup>‡</sup>, MADELEINE UDELL<sup>§</sup>, AND VOLKAN CEVHER<sup>‡</sup>

**Abstract.** This paper describes new algorithms for constructing a low-rank approximation of an input matrix from a *sketch*, a random low-dimensional linear image of the matrix. These algorithms come with rigorous performance guarantees. Empirically, the proposed methods achieve significantly smaller relative errors than other approaches that have appeared in the literature. For a concrete application, the paper outlines how the algorithms support on-the-fly compression of data from a direct Navier–Stokes (DNS) simulation.

**Key words.** Dimension reduction; matrix approximation; numerical linear algebra; randomized algorithm; single-pass algorithm; sketching; streaming algorithm; subspace embedding.

**AMS subject classifications.** Primary, 65F30; Secondary, 68W20.

**1. Motivation.** A *sketch* is a compressed data representation that supports updates to the underlying data and provides approximate answers to queries about the data. Over the last decade, sketches have emerged as a powerful tool for large-scale numerical linear algebra [51, 13, 25, 32, 50]. In particular, we can use a sketch to track a matrix that is presented as a sequence of linear updates, and we can extract a low-rank approximation of the induced matrix from the sketch. See [9, 21, 46, 45] for some recent work.

The purpose of this paper is to develop a new sketching method for low-rank matrix approximation in the streaming data model (section 2). We provide an informative mathematical analysis that explains the behavior of our algorithm (section 5). We also discuss implementation issues (section 4), and we present extensive numerical experiments on real and simulated data (section 6). The empirical performance of our technique is significantly better than earlier approaches (see subsection 6.1) that apply in the same setting.

Sketching methods for low-rank matrix approximation have many compelling applications. For instance, we have used these ideas to develop a storage-optimal algorithm for convex low-rank matrix optimization [52]. As a motivating example for this paper, we explain how sketching allows us to perform on-the-fly compression of data generated by large-scale computer simulations.

**1.1. Vignette: On-the-Fly Compression for Simulation.** Computer simulations often produce data matrices that are too large to store, process, or transmit in full. This challenge arises in a wide range of areas, including weather and climate forecasting [49, 17, 4], heat transfer and fluid flow [40, 6], computational fluid dynamics [5, 20], and aircraft design [36, 42]. Nevertheless, in these settings, the data matrix

---

\*First draft: 22 March 2017. First release: 16 July 2018.

**Funding:** JAT was supported in part by ONR Awards N00014-11-1002, N00014-17-1-214, N00014-17-1-2146, and the Gordon & Betty Moore Foundation. MU was supported in part by DARPA Award FA8750-17-2-0101. VC has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under the grant agreement number 725594 (time-data) and the Swiss National Science Foundation (SNSF) under the grant number 200021\_178865.

<sup>†</sup>California Institute of Technology, Pasadena, CA (jtropp@cms.caltech.edu).

<sup>‡</sup>École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (alp.yurtsever@epfl.ch, volkan.cevher@epfl.ch).

<sup>§</sup>Cornell University, Ithaca, NY (udell@cornell.edu).

often admits a good low-rank approximation. For many downstream applications, the low-rank approximation serves as well as—or even better than—the full data matrix because the approximation exposes latent structure [43, 11]. This observation raises the question of how to construct a low-rank approximation of simulation data efficiently.

We can model a simulation as a process that computes the state  $\mathbf{a}_{t+1} \in \mathbb{R}^m$  of a system at time  $t + 1$  from the state  $\mathbf{a}_t \in \mathbb{R}^m$  of the system at time  $t$ . The dimension  $m$  of the state increases with the resolution of the simulation. We may collect the data generated by the simulation into a matrix  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ .

The standard computational practice is to compute the full matrix  $\mathbf{A}$  and then to compress it. Methods include direct computation of a low-rank matrix or tensor approximation [53, 3] or fitting a statistical model [12, 23, 33]. These approaches usually involve storage costs of  $O(mn)$ .

In contrast, we consider replacing these techniques by a sketching algorithm. As each new state is computed, we update the sketch to reflect the arrival of a new column  $\mathbf{a}_t$  of the data matrix  $\mathbf{A}$ . Then we discard the state  $\mathbf{a}_t$ . Once the simulation is complete, we can extract a provably good rank- $r$  approximation of  $\mathbf{A}$  from the sketch. As we will see, this approach succeeds using total storage  $O(r(m + n))$ . For large matrices, the savings can be substantial. Subsection 6.7 contains a numerical demonstration of this idea.

**1.2. Summary of Related Work.** Randomized algorithms for low-rank matrix approximation were proposed in the theoretical computer science (TCS) literature in the late 1990s [39, 19]. Soon after, numerical analysts developed practical versions of these algorithms [34, 51, 41, 25, 24]. For more background on the history of randomized linear algebra, see [25, 32, 50].

Sketching algorithms are specifically designed for the streaming model; that is, for data that is presented as a sequence of updates. The paper [51] contains the first algorithm for low-rank approximation that can operate in this setting. The first explicit treatment of numerical linear algebra in the streaming model appears in [13]. Recent papers on low-rank matrix approximation in the streaming model include [9, 21, 46, 45]. We refer the reader to the latter works for additional background and information. This paper also includes detailed citations throughout.

**1.3. Notation.** We write  $\mathbb{F}$  for the scalar field, which is either real  $\mathbb{R}$  or complex  $\mathbb{C}$ . The symbol  $*$  refers to the (conjugate) transpose of a matrix or vector. The dagger  $\dagger$  denotes the Moore–Penrose pseudoinverse. We write  $\|\cdot\|_p$  for the Schatten  $p$ -norm for  $p \in [1, \infty]$ . The operator  $[\cdot]_r$  returns a (simultaneous) best rank- $r$  approximation of its argument with respect to the Schatten  $p$ -norms.

**2. Sketching and Low-Rank Approximation of a Matrix.** In this section, we describe the basic procedure for sketching a matrix and for computing a low-rank approximation from the sketch. We postpone the discussion of implementation details and variants to section 4.

**2.1. Dimension Reduction Maps.** We will use dimension reduction to collect information about an input matrix. Assume that  $k \leq n$ . A *randomized linear dimension reduction map* is a random matrix  $\Xi \in \mathbb{F}^{k \times n}$  with the property that

$$(2.1) \quad \mathbb{E} \|\Xi \mathbf{u}\|^2 = \text{const} \cdot \|\mathbf{u}\|^2 \quad \text{for all } \mathbf{u} \in \mathbb{F}^n.$$

In other words, the map reduces a vector of dimension  $n$  to dimension  $k$ , but it still preserves distances on average. It is also desirable that we can store the map  $\Xi$  and

84 apply it to vectors efficiently. See [section 3](#) for several concrete examples.

85 *Remark 2.1* (Geometry). The analysis of algorithms that use randomized dimen-  
86 sion reduction often depends on more detailed properties than the embedding condi-  
87 tion [\(2.1\)](#). See [\[25, 50\]](#) for more discussion.

88 **2.2. The Input Matrix.** Let  $\mathbf{A} \in \mathbb{F}^{m \times n}$  be an arbitrary matrix that we wish  
89 to approximate. In many applications where sketching is appropriate, the matrix is  
90 presented implicitly as a sequence of linear updates; see [subsection 2.4](#).

91 To apply sketching methods for low-rank matrix approximation, the user specifies  
92 a value  $r$  for the target rank of the approximation. The target rank  $r$  is typically far  
93 smaller than the smaller dimension  $\min\{m, n\}$  of the matrix.

**2.3. The Sketch.** Let us describe the sketching method we propose to acquire  
data about the input matrix. The sketch is parameterized by two natural numbers  
 $k, s$  that satisfy

$$r \leq k \leq s \leq \min\{m, n\},$$

94 where  $r$  is the target rank. In [subsection 5.6](#), we offer specific parameter recommen-  
95 dations that are supported by theoretical analysis. In [subsection 6.5](#), we demonstrate  
96 that these parameter choices are effective in practice.

97 Independently, draw and fix four randomized linear dimension reduction maps:

$$\begin{aligned} \Upsilon &\in \mathbb{F}^{k \times m} & \text{and} & & \Omega &\in \mathbb{F}^{k \times n}; \\ \Phi &\in \mathbb{F}^{s \times m} & \text{and} & & \Psi &\in \mathbb{F}^{s \times n}. \end{aligned} \tag{2.2}$$

99 The sketch itself consists of three matrices:

$$\mathbf{X} := \Upsilon \mathbf{A} \in \mathbb{F}^{k \times n} \quad \text{and} \quad \mathbf{Y} := \mathbf{A} \Omega^* \in \mathbb{F}^{m \times k}; \tag{2.3}$$

$$\mathbf{Z} := \Phi \mathbf{A} \Psi^* \in \mathbb{F}^{s \times s}. \tag{2.4}$$

103 The first two matrices  $(\mathbf{X}, \mathbf{Y})$  capture information about the co-range and the range  
104 of  $\mathbf{A}$ . The third matrix  $(\mathbf{Z})$  contains information about the action of  $\mathbf{A}$ .

105 *Remark 2.2* (Prior Work). The paper [\[48, Sec. 3\]](#) uses a sketch of the form [\(2.3\)](#)  
106 and [\(2.4\)](#) for low-rank matrix approximation. Related (but distinct) sketches appear  
107 in the papers [\[51, 13, 25, 50, 16, 10, 47, 46\]](#).

108 **2.4. Linear Updates.** In streaming data applications, the input matrix  $\mathbf{A} \in$   
109  $\mathbb{F}^{m \times n}$  is presented as a sequence of linear updates of the form

$$\mathbf{A} \leftarrow \theta \mathbf{A} + \tau \mathbf{H} \tag{2.5}$$

111 where  $\theta, \tau \in \mathbb{F}$  and the matrix  $\mathbf{H} \in \mathbb{F}^{m \times n}$ .

112 In view of the construction [\(2.3\)](#) and [\(2.4\)](#), we can update the sketch  $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$   
113 of the matrix  $\mathbf{A}$  to reflect the innovation [\(2.5\)](#) by means of the formulae

$$\begin{aligned} \mathbf{X} &\leftarrow \theta \mathbf{X} + \tau \Upsilon \mathbf{H} \\ \mathbf{Y} &\leftarrow \theta \mathbf{Y} + \tau \mathbf{H} \Omega^* \\ \mathbf{Z} &\leftarrow \theta \mathbf{Z} + \tau \Phi \mathbf{H} \Psi^*. \end{aligned} \tag{2.6}$$

115 *Remark 2.3* (Streaming Model). For the linear update model [\(2.5\)](#), randomized  
116 linear sketches are more or less the only way to track the input matrix [\[30\]](#). There  
117 are more restrictive streaming models (e.g., the columns of the matrix are presented  
118 in sequence) where it is possible to design other types of algorithms [\[18, 21\]](#).

119 **2.5. Computing a Low-Rank Approximation.** Once we have acquired a  
 120 sketch  $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$  of an input matrix  $\mathbf{A}$ , our goal is to produce a low-rank approxi-  
 121 mation. Let us outline the computations we propose. The intuition appears below in  
 122 [subsection 2.6](#), and [Section 5](#) presents the theoretical analysis.

123 The first two components  $(\mathbf{X}, \mathbf{Y})$  of the sketch are used to estimate the co-range  
 124 and the range of the matrix  $\mathbf{A}$ . Compute thin orthogonal–triangular factorizations:

$$125 \quad (2.7) \quad \begin{aligned} \mathbf{X}^* &=: \mathbf{P}\mathbf{R}_1 \quad \text{where } \mathbf{P} \in \mathbb{F}^{n \times k}; \\ \mathbf{Y} &=: \mathbf{Q}\mathbf{R}_2 \quad \text{where } \mathbf{Q} \in \mathbb{F}^{m \times k}. \end{aligned}$$

126 Both  $\mathbf{P}$  and  $\mathbf{Q}$  have orthonormal columns; we discard the triangular parts  $\mathbf{R}_1$  and  
 127  $\mathbf{R}_2$ . The third sketch  $\mathbf{Z}$  is used to compute the core matrix  $\mathbf{W}$ , which describes the  
 128 predominant action of the matrix:

$$129 \quad (2.8) \quad \mathbf{W} := (\Phi\mathbf{Q})^\dagger \mathbf{Z} ((\Psi\mathbf{P})^\dagger)^* \in \mathbb{F}^{k \times k}.$$

130 Last, we construct a rank- $k$  approximation  $\hat{\mathbf{A}}$  of the input matrix  $\mathbf{A}$ :

$$131 \quad (2.9) \quad \hat{\mathbf{A}} := \mathbf{Q}\mathbf{W}\mathbf{P}^*$$

132 In some situations, it is more desirable to produce an approximation with exact rank  
 133  $r$ . To do so, we simply replace  $\hat{\mathbf{A}}$  by its best rank- $r$  approximation:

$$134 \quad (2.10) \quad \llbracket \hat{\mathbf{A}} \rrbracket_r = \mathbf{Q} \llbracket \mathbf{W} \rrbracket_r \mathbf{P}^*.$$

135 [The formula (2.10) is an easy consequence of the Eckart–Young Theorem [26, Sec. 6]  
 136 and the fact that  $\mathbf{Q}, \mathbf{P}$  have orthonormal columns.]

137 *Remark 2.4 (Extensions).* We can construct other structured approximations of  
 138  $\mathbf{A}$  by projecting  $\hat{\mathbf{A}}$  onto a set of structured matrices. See [46, Secs. 5–6] for a discussion  
 139 of this idea in the context of another sketching technique. See our paper [45] for a  
 140 sketching method designed for positive-semidefinite matrices.

141 *Remark 2.5 (Prior Work).* The reconstruction formulae (2.9) and (2.10) are new.  
 142 The papers [51, 13, 25, 50, 16, 10, 48, 47, 46] describe alternative methods for low-rank  
 143 matrix approximation from a sketch. The numerical work in [section 6](#) demonstrates  
 144 that the performance of our method is uniformly superior to the earlier techniques.

145 **2.6. Intuition.** The low-rank approximations (2.9) and (2.10) are based on some  
 146 well-known insights from randomized linear algebra [25, Sec. 1]. Since  $\mathbf{P}$  and  $\mathbf{Q}$   
 147 capture the co-range and range of the input matrix, we expect that

$$148 \quad (2.11) \quad \mathbf{A} \approx \mathbf{Q}(\mathbf{Q}^* \mathbf{A} \mathbf{P}) \mathbf{P}^*$$

(See [Lemma SM1.5](#) for justification.) We cannot compute the core matrix  $\mathbf{Q}^* \mathbf{A} \mathbf{P}$   
 directly from a linear sketch because  $\mathbf{P}$  and  $\mathbf{Q}$  are functions of  $\mathbf{A}$ . Even so, we can  
 estimate the core matrix using the action sketch  $\mathbf{Z}$ . Observe that

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

The approximation (2.11) allows us to drop the second term, so

$$\mathbf{Z} \approx (\Phi\mathbf{Q})(\mathbf{Q}^* \mathbf{A} \mathbf{P})(\mathbf{P}^* \Psi^*).$$

149 Transfer the outer matrices to the left-hand side to discover that

150 (2.12) 
$$\mathbf{W} = (\Phi\mathbf{Q})^\dagger \mathbf{Z}((\Psi\mathbf{P})^\dagger)^* \approx \mathbf{Q}^* \mathbf{A} \mathbf{P}.$$

In view of (2.11) and (2.12), we arrive at

$$\mathbf{A} \approx \mathbf{Q}(\mathbf{Q}^* \mathbf{A} \mathbf{P}) \mathbf{P}^* \approx \mathbf{Q} \mathbf{W} \mathbf{P}^* = \hat{\mathbf{A}}.$$

When  $\hat{\mathbf{A}}$  is a good approximation of  $\mathbf{A}$ , we can project it onto the set of rank- $r$  matrices without increasing the error substantially:

$$\mathbf{A} \approx \llbracket \hat{\mathbf{A}} \rrbracket_r = \mathbf{Q} \llbracket \mathbf{W} \rrbracket_r \mathbf{P}^*.$$

151 **Theorem 5.1** and **Corollary 5.3** justify these heuristics completely for Gaussian di-  
 152 mension reduction maps.

153 *Remark 2.6* (Prior Work). Our method is inspired by the intuition in [25, Sec. 1],  
 154 which also motivates the low-rank sketching algorithms in [47, 46]. The sketching  
 155 techniques in the TCS literature [13, 50, 16, 10, 48] are based on a different idea.

156 **3. Randomized Linear Dimension Reduction Maps.** In this section, we  
 157 describe several randomized linear dimension reduction maps that are suitable for  
 158 implementing sketching algorithms for low-rank matrix approximation. See [31, 25,  
 159 50, 46] for additional discussion and examples.

160 **3.1. Gaussian Maps.** The most basic dimension reduction map is simply a  
 161 Gaussian matrix. That is,  $\Xi \in \mathbb{F}^{k \times n}$  is a  $k \times n$  matrix with independent standard  
 162 normal entries.<sup>1</sup>

163 **Algorithm SM3.6** describes an implementation of Gaussian dimension reduction.  
 164 The map  $\Xi$  requires storage of  $kn$  floating-point numbers in the field  $\mathbb{F}$ . The cost of  
 165 applying the map to a vector is  $O(kn)$  arithmetic operations.

166 Gaussian dimension reduction maps are simple, and they are effective in random-  
 167 ized algorithms for low-rank matrix approximation [25]. We can also analyze their  
 168 behavior in full detail; see section 5. On the other hand, it is expensive to draw a  
 169 large number of Gaussian random variables, and the cost of storage and arithmetic  
 170 renders these maps less appealing for sketching applications.

171 *Remark 3.1* (History). Gaussian dimension reduction has been used as an algo-  
 172 rithmic tool since the paper of Indyk & Motwani [28]. In spirit, this approach is  
 173 quite similar to the earlier theoretical work of Johnson & Lindenstrauss [29], which  
 174 performs dimension reduction by projection onto a random subspace.

175 **3.2. Scrambled SRFT Maps.** Next, we describe a structured dimension re-  
 176 duction map, called a *scrambled subsampled randomized Fourier transform* (SSRFT).  
 177 We recommend this approach for practical implementations.

An SSRFT map takes the form

$$\Xi = \mathbf{R} \mathbf{F} \mathbf{\Pi} \mathbf{\Pi}' \in \mathbb{F}^{k \times n}.$$

178 The matrices  $\mathbf{\Pi}, \mathbf{\Pi}' \in \mathbb{F}^{n \times n}$  are signed permutations,<sup>2</sup> drawn independently and  
 179 uniformly at random. The matrix  $\mathbf{F} \in \mathbb{F}^{n \times n}$  denotes a discrete cosine transform

---

<sup>1</sup>A real standard normal variable follows the Gaussian distribution with mean zero and variance one. A complex standard normal variable takes the form  $g_1 + ig_2$ , where  $g_i$  are independent real standard normal variables.

<sup>2</sup>A signed permutation matrix has precisely one nonzero entry in each row and column, and each nonzero entry of the matrix has modulus one.

180 ( $\mathbb{F} = \mathbb{R}$ ) or a discrete Fourier transform ( $\mathbb{F} = \mathbb{C}$ ). The matrix  $\mathbf{R} \in \mathbb{F}^{k \times n}$  is a restriction  
 181 to  $k$  coordinates, chosen uniformly at random.

182 **Algorithm SM3.7** presents an implementation of an SSRFT. The cost of storing  
 183  $\mathbf{\Xi}$  is just  $O(n)$  numbers. The cost of applying  $\mathbf{\Xi}$  to a vector is  $O(n \log n)$  arithmetic  
 184 operations, using the Fast Fourier Transform (FFT) or the Fast Cosine Transform  
 185 (FCT). This cost can be reduced [51] further to  $O(n \log k)$ , but the improvement is  
 186 rarely worth the implementation effort.

187 In practice, SSRFTs behave almost the same way as Gaussian matrices, but their  
 188 storage cost does not scale with the output dimension  $k$ . On the other hand, the  
 189 analysis [2, 44, 8] is less complete than in the Gaussian case [25]. A proper implemen-  
 190 tation requires fast trigonometric transforms. Last, the random permutations and  
 191 FFTs require data movement, which could be a challenge in the distributed setting.

192 *Remark 3.2 (History).* SSRFTs are inspired by the work of Ailon & Chazelle [2] on  
 193 fast Johnson–Lindstrauss transforms. For applications in randomized linear algebra,  
 194 see the papers [51, 31, 25, 44, 8].

195 **3.3. Sparse Sign Matrices.** Last, we describe another type of randomized  
 196 dimension reduction map, called a *sparse sign matrix*. We recommend these maps for  
 197 practical implementations where data movement (i.e., coherency) is a concern.

198 To construct a sparse sign matrix  $\mathbf{\Xi} \in \mathbb{F}^{k \times n}$ , we fix a sparsity parameter  $\zeta$  in the  
 199 range  $2 \leq \zeta \leq k$ . The columns of the matrix are drawn independently at random.  
 200 To construct each column, we take  $\zeta$  iid draws from the  $\text{UNIFORM}\{z \in \mathbb{F} : |z| = 1\}$   
 201 distribution, and we place these random variables in  $p$  coordinates, chosen uniformly  
 202 at random. Empirically, we have found that  $\zeta = \min\{k, 2 \log(1 + n)\}$  is an effective  
 203 parameter selection. See [15] for some theoretical justification.

204 **Algorithm SM3.8** describes an implementation of sparse dimension reduction.  
 205 Since the matrix  $\mathbf{\Xi} \in \mathbb{F}^{k \times n}$  has  $\zeta$  nonzeros per column, we can store the matrix with  
 206  $O(\zeta n \log(1 + k/\zeta))$  numbers. The cost of applying the map to a vector is  $O(\zeta n)$   
 207 arithmetic operations.

208 Sparse sign matrices have benefits for data coherency because the columns are  
 209 generated independently and the matrices can be applied using (blocked) matrix mul-  
 210 tiplication. One weakness is that we must use sparse data structures and arithmetic  
 211 to enjoy the benefit of these maps.

212 *Remark 3.3 (History).* Sparse dimension reduction maps are inspired by the work  
 213 of Achlioptas [1] on database-friendly random projections. For applications in ran-  
 214 domized linear algebra, see [14, 35, 37, 38, 7, 15].

215 **4. Implementation and Costs.** This section contains further details about the  
 216 implementation of the sketching and reconstruction methods from [section 2](#), including  
 217 an account of storage and arithmetic costs. All pseudocode appears in [section SM2](#).  
 218 The supplementary materials include MATLAB code for the algorithms.

219 **4.1. Sketching and Updates.** [Algorithms SM3.1](#) and [SM3.2](#) contain the pseu-  
 220 docode for initializing the sketch and for performing the linear update (2.5).

221 The sketch requires the storage of four dimension reduction maps with size  $k \times m$ ,  
 222  $k \times n$ ,  $s \times m$ ,  $s \times n$ . We recommend using SSRFTs or sparse sign matrices to minimize  
 223 the storage costs associated with the dimension reduction maps.

224 The sketch itself consists of three matrices with dimensions  $k \times n$ ,  $m \times k$ , and  $s \times s$ .  
 225 In general, the sketch matrices are dense, so they require  $k(m + n) + s^2$  floating-point  
 226 numbers in the field  $\mathbb{F}$ .

227 The arithmetic cost of the linear update  $\mathbf{A} \leftarrow \theta \mathbf{A} + \tau \mathbf{H}$  is dominated by the

228 minimum cost of computing  $\Phi H$  or  $H\Psi$ . That is, we apply the dimension reduction  
 229 map to  $s$  vectors of length  $\min\{m, n\}$ . The cost of the update depends heavily on the  
 230 structure of the matrix  $H$  and the type of dimension reduction map.

231 **4.2. Low-Rank Approximation.** Algorithm SM3.3 lists the pseudocode for  
 232 computing a rank- $k$  approximation  $\hat{A}$  of the matrix  $A$  contained in the sketch;  
 233 see (2.9).

234 The method requires additional storage of  $k(m+n)$  numbers for the orthonormal  
 235 matrices  $P$  and  $Q$ , as well as  $k^2$  numbers for the core matrix  $W$ . The arithmetic cost  
 236 is usually dominated by the computation of the orthogonal–triangular factorizations  
 237 of  $X^*$  and  $Y$ , which require  $O(k^2(m+n))$  operations. When the parameters satisfy  
 238  $s \gg k$ , it is possible that the cost  $O(ks^2)$  of forming the core matrix  $W$  will be larger.

239 **4.3. Fixed-Rank Approximation.** Algorithm SM3.4 presents the pseudocode  
 240 for computing the rank- $r$  approximation  $[[\hat{A}]]_r$  of the matrix  $A$  contained in the sketch;  
 241 see (2.10).

242 The working storage cost  $O(k(m+n))$  is dominated by the call to the routine  
 243 Algorithm SM3.3. Typically, the arithmetic cost is also dominated by the  $O(k^2(m+n))$   
 244 cost of the call to Algorithm SM3.3. When  $s \gg k$ , it is possible that the  $O(s^3)$   
 245 cost of the truncated SVD will drive the arithmetic cost.

246 **5. Theoretical Results.** It is always important to characterize the behavior of  
 247 numerical algorithms, but the challenge is more acute for sketching methods. Indeed,  
 248 we cannot store the stream of updates, so we cannot repeat the computation with  
 249 new parameters if it is unsuccessful. As a consequence, we must perform *a priori*  
 250 theoretical analysis to be able to implement sketching algorithms with confidence.

251 In this section, we analyze our sketching and reconstruction algorithms in the  
 252 ideal case where all of the dimension reduction maps are standard normal. These  
 253 results allow us to make concrete recommendations for the sketch size parameters.  
 254 Empirically, other types of dimension reduction exhibit the identical performance  
 255 (subsection 6.4), so our analysis also supports more practical implementations based  
 256 on SSRFTs or sparse sign matrices. The numerical work in section 6 confirms the  
 257 value of this analysis.

**5.1. The Tail Energy.** For each natural number  $r$ , define the  $r$ th *tail energy* of  
 the input matrix

$$\tau_r^2(A) := \min_{\text{rank } B < r} \|A - B\|_2^2 = \sum_{j \geq r} \sigma_j^2(A),$$

258 where  $\sigma_j$  returns the  $j$ th largest singular value of a matrix. The second identity  
 259 follows from the Eckart–Young Theorem [26, Sec. 6].

260 **5.2. The Field Parameter.** We also introduce a parameter that reflects the  
 261 field over which we are working:

$$(5.1) \quad \alpha := \alpha(\mathbb{F}) := \begin{cases} 1, & \mathbb{F} = \mathbb{R} \\ 0, & \mathbb{F} = \mathbb{C}. \end{cases}$$

263 This quantity allows us to capture the behavior of real and complex Gaussian matrices  
 264 within the same formula.

265 **5.3. Analysis of Low-Rank Approximation.** The first result gives a bound  
 266 for the expected error in the rank- $k$  approximation  $\hat{A}$  of the input matrix  $A$ .

267 **THEOREM 5.1** (Low-Rank Approximation: Error Bound). *Let  $\mathbf{A} \in \mathbb{F}^{m \times n}$  be*  
 268 *an arbitrary input matrix. Assume that the sketch size parameters satisfy  $s \geq 2k +$*   
 269  *$\alpha$ . Draw independent Gaussian dimension reduction maps  $(\Upsilon, \Omega, \Phi, \Psi)$ , as in (2.2).*  
 270 *Extract a sketch (2.3) and (2.4) of the input matrix. Then the rank- $k$  approximation*  
 271  *$\hat{\mathbf{A}}$ , constructed in (2.9), satisfies the error bound*

$$272 \quad (5.2) \quad \mathbb{E} \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 \leq \frac{s - \alpha}{s - k - \alpha} \cdot \min_{\varrho < k - \alpha} \frac{k + \varrho - \alpha}{k - \varrho - \alpha} \cdot \tau_{\varrho+1}^2(\mathbf{A}).$$

273 We postpone the proof to [section SM1](#). The analysis is similar in spirit to the proof  
 274 of [46, Thm. 4.3], but it is somewhat more challenging.

275 [Theorem 5.1](#) contains explicit and reasonable constants, so we can use it to design  
 276 algorithms that achieve a specific error tolerance. For example, suppose that  $r$  is the  
 277 target rank of the approximation. Then the choice

$$278 \quad (5.3) \quad k = 5r + \alpha \quad \text{and} \quad s = 2k + \alpha$$

ensures that the error in the rank- $k$  approximation  $\hat{\mathbf{A}}$  is within a constant factor 3 of  
 the optimal rank- $r$  approximation:

$$\mathbb{E} \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 \leq 3 \cdot \tau_{r+1}^2(\mathbf{A}).$$

279 In practice, we have found the parameter selection (5.3) to be effective for a range  
 280 of examples. Moreover, if  $k/r \rightarrow \infty$  and  $s/k \rightarrow \infty$ , we drive the leading constant in  
 281 (5.2) to one.

282 The true meaning of [Theorem 5.1](#) is more subtle. The minimum over  $\varrho$  indicates  
 283 that the approximation automatically adapts to the spectral decay of the input matrix.  
 284 This effect is usually more significant than any benefit we may achieve by adjusting  
 285 the parameters to control the leading constant. In [subsection 5.6](#), we exploit this idea  
 286 to recommend sketch size parameters for a given storage budget.

287 *Remark 5.2* (Failure probability). It is well known that the expected performance  
 288 of randomized linear algebra methods also characterizes the typical performance [25,  
 289 Fig. 7.3]. The probability that the error is significantly larger than (5.2) is negligible.

290 **5.4. Analysis of Fixed-Rank Approximation.** Our second result gives a  
 291 bound for the error in the rank- $r$  approximation  $\llbracket \hat{\mathbf{A}} \rrbracket_r$  of the input matrix  $\mathbf{A}$ .

**COROLLARY 5.3** (Fixed-Rank Approximation: Error Bound). *Instate the as-*  
*sumptions of [Theorem 5.1](#). Then the rank- $r$  approximation  $\llbracket \hat{\mathbf{A}} \rrbracket_r$  satisfies the error*  
*bound*

$$\mathbb{E} \|\mathbf{A} - \llbracket \hat{\mathbf{A}} \rrbracket_r\|_2 \leq \tau_{r+1}(\mathbf{A}) + 2 \left[ \frac{s - \alpha}{s - k - \alpha} \cdot \min_{\varrho < k - \alpha} \frac{k + \varrho - \alpha}{k - \varrho - \alpha} \cdot \tau_{\varrho+1}^2(\mathbf{A}) \right]^{1/2}.$$

292 This statement is an immediate consequence of [Theorem 5.1](#) and the result [46,  
 293 Prop. 6.1]. We omit the details.

294 Let us elaborate on [Corollary 5.3](#). When the approximation  $\hat{\mathbf{A}}$  is a good rank- $k$   
 295 approximation of  $\mathbf{A}$ , then the matrix  $\llbracket \hat{\mathbf{A}} \rrbracket_r$  is also a good rank- $r$  approximation of  
 296  $\mathbf{A}$ . In particular, the rank- $r$  approximation can exploit decay in the spectrum of the  
 297 input matrix. The empirical work in [section 6](#) highlights the practical importance of  
 298 this phenomenon.

299 **5.5. The Storage Budget.** It is important to understand the storage we need  
 300 to maintain a sketch of an input matrix. We have recommended using structured  
 301 dimension reduction maps  $(\mathbf{Y}, \mathbf{\Omega}, \mathbf{\Phi}, \mathbf{\Psi})$  so the storage cost for the dimension reduction  
 302 maps does not increase with the sketch size parameters  $(k, s)$ . In this case, we may  
 303 focus on the cost of maintaining the sketch  $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$  itself.

304 Counting dimensions, via (2.3) and (2.4), we see that the three sketch matrices  
 305 require a total storage budget of

$$306 \quad (5.4) \quad T := k(m + n) + s^2$$

floating-point numbers in the field  $\mathbb{F}$ . To achieve a rank- $r$  approximation, the minimum allowable values for the sketch size parameters are  $k_{\min} = r + \alpha + 1$  and  $s_{\min} = 2k_{\min} + \alpha$ . Therefore, the minimum storage budget is

$$T_{\min}(r) := (r + \alpha + 1)(m + n) + (2r + 3\alpha + 1)^2.$$

307 Of course, larger parameters  $(k, s)$  support better approximations. In the next section,  
 308 we offer a more practical approach for choosing  $(k, s)$ .

309 **5.6. Theoretical Guidance for Sketch Size Parameters.** Suppose that we  
 310 fix the storage budget  $T$ , defined in (5.4). We may ask how to apportion the sketch  
 311 size parameters  $(k, s)$  to achieve superior empirical performance. Theorem 5.1 offers  
 312 insight on this question; see subsection 6.5 for numerical support.

313 **5.6.1. General Spectrum.** To control the theoretical bound Theorem 5.1 on  
 314 the approximation error, it is natural to make the parameter  $k$  as large as possible.  
 315 Indeed, when  $k$  is large, the parameter  $\rho$  in the error bound (5.2) has more room to  
 316 adapt to decay in the spectrum of  $\mathbf{A}$ . Note that the condition  $s \geq 2k + \alpha$  ensures  
 317 that the first fraction in the error bound cannot exceed two.

318 Therefore, for  $T \geq T_{\min}(r)$ , we pose the optimization problem

$$319 \quad (5.5) \quad \max k \quad \text{subject to} \quad s \geq 2k + \alpha, \quad \text{and} \quad k(m + n) + s^2 = T.$$

320 Up to rounding, the solution is

$$321 \quad (5.6) \quad \begin{aligned} k_{\dagger} &:= \left\lfloor \frac{1}{8} \left( \sqrt{(m + n + 4\alpha)^2 + 16(T - \alpha^2)} - (m + n + 4\alpha) \right) \right\rfloor; \\ s_{\dagger} &:= \left\lfloor \sqrt{T - k_{\dagger}(m + n)} \right\rfloor. \end{aligned}$$

322 The parameter choice  $(k_{\dagger}, s_{\dagger})$  is suitable for a wide range of examples.

323 **5.6.2. Flat Spectrum.** Suppose we know that the spectrum of the input matrix  
 324 does not decay past a certain point:  $\sigma_j(\mathbf{A}) \approx \sigma_{\hat{r}}(\mathbf{A})$  for  $j > \hat{r}$ . In this case, the  
 325 minimum value of the error (5.2) tends to occur when  $\rho = \hat{r}$ .

326 In this case, we can obtain a theoretically supported parameter choice  $(k_b, s_b)$  by  
 327 numerical solution of the optimization problem

$$328 \quad (5.7) \quad \min \frac{s - \alpha}{s - k - \alpha} \cdot \frac{k + \hat{r} - \alpha}{k - \hat{r} - \alpha} \quad \text{subject to} \quad s \geq 2k + \alpha, \quad k \geq \hat{r} + \alpha + 1, \\ \text{and} \quad k(m + n) + s^2 = T.$$

329 In fact, this problem admits a closed-form solution, but we have chosen to omit the  
 330 complicated formula.

331 **6. Numerical Experiments.** This section presents computer experiments that  
 332 are designed to evaluate the performance of the proposed sketching algorithms for low-  
 333 rank matrix approximation. We include comparisons with alternative methods from  
 334 the literature, and we argue that the proposed approach produces superior results.

335 **6.1. Alternative Sketching and Reconstruction Methods.** We compare  
 336 our approach with two sketching algorithms for low-rank matrix approximation that  
 337 have appeared in the literature. Our recent work [46] identifies these two algorithms  
 338 as the best techniques available, so we omit comparisons with additional methods.

339 **6.1.1. A Three-Sketch Method.** Boutsidis et al. [10, Sec. 6] recently intro-  
 340 duced a new method for low-rank matrix approximation from a sketch; Upadhyay [48,  
 341 Sec. 3] later proposed some refinements.

Upadhyay’s variant is based on the same kind of sketch (2.2)–(2.4) that we are  
 using in this paper. He develops the following formula for approximating the input  
 matrix. First, compute orthonormal bases  $\mathbf{Q}$  and  $\mathbf{P}$  for the range and co-range  
 via (2.7). Then form thin singular value decompositions:

$$\Phi\mathbf{Q} = \mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^* \quad \text{and} \quad \Psi\mathbf{P} = \mathbf{U}_2\mathbf{S}_2\mathbf{V}_2^*.$$

342 Construct the rank- $r$  approximation

$$343 \quad (6.1) \quad \hat{\mathbf{A}}_{\text{upa}} = \mathbf{Q}\mathbf{V}_1\mathbf{S}_1^\dagger \llbracket \mathbf{U}_1^*\mathbf{Z}\mathbf{U}_2 \rrbracket_r \mathbf{S}_2^\dagger\mathbf{V}_2^*\mathbf{P}^*.$$

344 Superficially, the approximation  $\hat{\mathbf{A}}_{\text{upa}}$  may look similar to the approximation we de-  
 345 veloped in (2.10). Nevertheless, they are designed using different principles, and their  
 346 performance is quite different in practice.

**6.1.2. A Two-Sketch Method.** In [46], we developed and analyzed a very  
 simple sketching algorithm for low-rank matrix approximation. This approach uses  
 only two dimension reduction maps:

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

The sketch takes the form

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

347 To obtain a rank- $r$  approximation from this sketch, we compute

$$348 \quad (6.2) \quad \mathbf{Y} = \mathbf{Q}\mathbf{R} \quad \text{and} \quad \hat{\mathbf{A}}_{\text{two}} = \mathbf{Q} \llbracket (\Upsilon\mathbf{Q})^\dagger \mathbf{X} \rrbracket_r.$$

349 The numerically stable implementation is a little more complicated; see [46, Alg. 7]  
 350 for details.

351 **6.2. Experimental Setup.** Our experimental design is quite similar to our  
 352 previous papers [46, 45] on sketching algorithms for low-rank matrix approximation.

353 **6.2.1. Procedure.** Fix an input matrix  $\mathbf{A} \in \mathbb{F}^{n \times n}$  and a target rank  $r$ . Then  
 354 select the sketch size parameters  $(k, s)$  or  $(k, \ell)$ . For each trial, we draw dimension  
 355 reduction maps from a specified distribution and form the sketch of the input matrix.  
 356 We compute a rank- $r$  approximation  $\hat{\mathbf{A}}_{\text{out}}$  using a specified reconstruction formula.  
 357 The approximation error is calculated relative to the best rank- $r$  approximation error  
 358 in Schatten  $p$ -norm:

$$359 \quad (6.3) \quad S_p \text{ relative error} = \frac{\|\mathbf{A} - \hat{\mathbf{A}}_{\text{out}}\|_p}{\|\mathbf{A} - \llbracket \mathbf{A} \rrbracket_r\|_p} - 1.$$

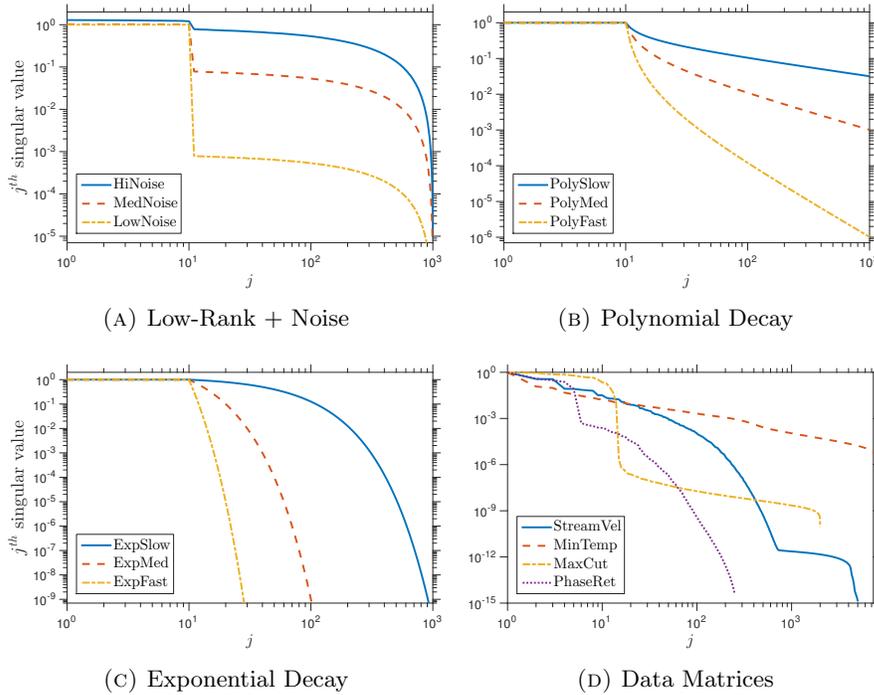


FIG. 1: **Spectra of input matrices.** Plots of the singular value spectrum for an input matrix from each of the synthetic classes (**LowRank**, **PolyDecay**, **ExpDecay** with effective rank  $R = 10$ ) and from each of the real data classes (**MinTemp**, **StreamVel**, **MaxCut**, **PhaseRetrieval**) described in subsection 6.3.

360 We perform 20 independent trials and report the average error. Owing the measure  
 361 concentration effects, the average error is also the typical error; cf. [25, Fig. 7.3].

362 The body of this paper presents a limited selection of results. Section SM3 con-  
 363 tains additional numerical evidence. The supplementary materials also include MAT-  
 364 LAB code that can reproduce these experiments.

365 **6.2.2. The Oracle Error.** To make fair comparisons among algorithms, we  
 366 fix the storage budget and identify the parameter choices that minimize the relative  
 367 error (6.3) incurred. We refer to the minimum as the *oracle error* for an algorithm.

368 For our new reconstruction (2.10) and for the Upadhyay method (6.1), we compute  
 369 the total storage cost as  $T = k(m+n) + s^2$  and we require that  $k > r + \alpha$  and  $s \geq 2k + \alpha$ .  
 370 For the two-sketch method (6.2), the total storage cost is  $T = km + \ell n$  and we require  
 371 that  $k > r + \alpha$  and  $\ell > k + \alpha$ . Note that the storage budget neglects the cost of storing  
 372 the dimension reduction maps because this cost has lower order than the sketch when  
 373 we use structured dimension reduction maps.

374 **6.3. Classes of Input Matrices.** As in our previous papers [46, 45], we consider  
 375 several different types of synthetic and real input matrices. See Figure 1 for a plot of  
 376 the spectra of these input matrices.

377 **6.3.1. Synthetic Examples.** We work over the complex field  $\mathbb{C}$ . The matrix di-  
 378 mensions  $m = n = 10^3$ , and we introduce an effective rank parameter  $R \in \{5, 10, 20\}$ .

379 We compute an approximation with actual rank  $r = 10$ .

1. **Low-rank + noise:** Let  $\xi \geq 0$  be a signal-to-noise parameter. These matrices take the form

$$\mathbf{A} = \text{diag}(\underbrace{1, \dots, 1}_R, 0, \dots, 0) + \xi n^{-1} \mathbf{W} \in \mathbb{C}^{n \times n},$$

380 where  $\mathbf{W} = \mathbf{G}\mathbf{G}^*$  for a standard normal matrix  $\mathbf{G} \in \mathbb{F}^{n \times n}$ . We consider  
 381 several parameter values: `LowRankLowNoise` ( $\xi = 10^{-4}$ ), `LowRankMedNoise`  
 382 ( $\xi = 10^{-2}$ ), `LowRankHiNoise` ( $\xi = 10^{-1}$ ).

2. **Polynomial decay:** For a decay parameter  $p > 0$ , consider matrices

$$\mathbf{A} = \text{diag}(\underbrace{1, \dots, 1}_R, 2^{-p}, 3^{-p}, \dots, (n - R + 1)^{-p}) \in \mathbb{C}^{n \times n}.$$

383 We study three examples: `PolyDecaySlow` ( $p = 0.5$ ), `PolyDecayMed` ( $p = 1$ ),  
 384 `PolyDecayFast` ( $p = 2$ ).

3. **Exponential decay:** For a decay parameter  $q > 0$ , consider matrices

$$\mathbf{A} = \text{diag}(\underbrace{1, \dots, 1}_R, 10^{-q}, 10^{-2q}, \dots, 10^{-(n-R)q}) \in \mathbb{C}^{n \times n}.$$

385 We consider the cases `ExpDecaySlow` ( $q = 0.01$ ), `ExpDecayMed` ( $q = 0.1$ ),  
 386 `ExpDecayFast` ( $q = 0.5$ ).

387 **6.3.2. Application Examples.** We also consider instances of low-rank data  
 388 matrices that arise in applications. For these matrices, we consider a range of values  
 389 for the actual rank  $r$  of the approximation.

- 390 1. **Navier–Stokes:** We test the hypothesis, discussed in [subsection 1.1](#), that  
 391 sketching methods can be used to perform on-the-fly compression of the out-  
 392 put of a PDE simulation. We have obtained a 2D Direct Navier–Stokes (DNS)  
 393 simulation of low-Reynolds number flow around a cylinder on a coarse mesh.  
 394 The simulation is started impulsively from a rest state. Transient dynamics  
 395 emerge in the first third of the simulation, while the remaining time steps  
 396 capture the limit cycle. Each of the velocity and pressure fields is centered  
 397 around its temporal mean. This data is courtesy of Beverley McKeon and  
 398 Sean Symon.

399 The real  $m \times n$  matrix `StreamVel` contains streamwise velocities at  $m =$   
 400  $10,738$  points for each of  $n = 5,001$  time instants. The first 20 singular  
 401 values of the matrix decay by two orders of magnitude, and the rest of the  
 402 spectrum exhibits slow exponential decay. This is typical for physical models.

- 403 2. **Weather:** We also test the hypothesis that sketching methods can be used  
 404 to perform on-the-fly compression of temporal data as it is collected. We have  
 405 obtained a matrix that tabulates meteorological variables at weather stations  
 406 across the northeastern United States on days during the years 1981–2016.  
 407 This data is courtesy of William North.

408 The real  $m \times n$  matrix `MinTemp` contains the minimum temperature recorded  
 409 at each of  $m = 19,264$  stations on each of  $n = 7,305$  days. The first 10 singu-  
 410 lar values decay by two orders of magnitude, while the rest of the spectrum  
 411 has medium polynomial decay. This is typical for measured data.

412 **3. Sketchy Decisions:** Last, we consider matrices that arise from an opti-  
 413 mization algorithm for solving large-scale semidefinite programs [52]. In this  
 414 application, the data matrices are presented as a long series of rank-one up-  
 415 dates, and sketching is a key element of the algorithm.

416 (a) `MaxCut`: This is a real psd matrix with  $m = n = 2,000$  that gives a  
 417 high-accuracy solution to the `MAXCUT` SDP for a sparse graph [22].  
 418 This matrix is effectively rank deficient with  $R = 14$ , and the spec-  
 419 trum has fast exponential decay after this point.

420 (b) `PhaseRetrieval`: This is a complex psd matrix with  $m = n = 25,921$   
 421 that gives a low-accuracy solution to a phase retrieval SDP [27]. This  
 422 matrix is effectively rank deficient with  $R = 5$ , and the spectrum has  
 423 fast exponential decay after this point.

424 **6.4. Insensitivity to Dimension Reduction Map.** Our first experiment is  
 425 designed to show that the proposed rank- $r$  reconstruction formula (2.10) is insensitive  
 426 to the distribution of the dimension reduction map at the oracle parameter values  
 427 (subsection 6.2.2).

428 Figure 2 presents experiments with synthetic test matrices with effective rank  
 429  $R = 10$ , approximation rank  $r = 10$ , and the Schatten 2-norm error (6.3). For most  
 430 storage budgets  $T$ , the Gaussian, SSRFT, and sparse dimension reduction maps yield  
 431 equivalent values for the oracle error. In fact, because it is unitary, the SSRFT map  
 432 even performs slightly *better* than the others when the storage budget is very large.  
 433 See subsection SM3.1 for more numerics, which transmit the same message.

434 The other reconstruction methods (6.1) and (6.2) are also insensitive to the choice  
 435 of dimension reduction maps. We omit the numerical evidence. These observations  
 436 justify the transfer of theoretical and empirical results for Gaussians to SSRFT and  
 437 sparse dimension reduction maps.

438 **6.5. Approaching the Oracle Performance.** Next, we show that theoretical  
 439 parameter choices in (2.10) produce results almost as good as the oracle performance.

440 Figures 3 and 4 display the outcome of the following experiment. For synthetic  
 441 test matrices with effective rank  $R = 10$  and approximation rank  $r = 10$ , we compare  
 442 the oracle performance (subsection 6.2.2) of our rank- $r$  approximation (2.10) with  
 443 its performance at the theoretical parameters proposed in subsection 5.6. (In the  
 444 formula (5.7) for a flat spectrum, we set the tail location  $\hat{r} = r$ .) We use Gaussian  
 445 dimension reduction maps, but equivalent results hold for other types of dimension  
 446 reduction maps. See subsection SM3.2 for effective rank  $R = 5$  and  $R = 20$ .

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

455 **6.6. Comparison of Reconstruction Formulas: Synthetic Examples.** Let  
 456 us now compare the proposed rank- $r$  reconstruction formula (2.10) with the Upadhyay  
 457 approximation (6.1) and the two-sketch approximation (6.2).

458 Figures 5 and 6 present the results of the following experiment. For synthetic  
 459 matrices with effective rank  $R = 10$  and approximation rank  $r = 10$ , we compare the  
 460 relative error (6.3) achieved by each of the three rank- $r$  reconstructions as a function  
 461 of storage (subsection 6.2.2). We use Gaussian dimension reduction maps in these  
 462 experiments; similar results are evident for other types of maps. Results for effective  
 463 rank  $R = 5$  and  $R = 20$  appear in subsection SM3.3.

464 Let us make some remarks:

- 465 • This experiment demonstrates clearly that the proposed approximation (2.10)  
 466 dominates the earlier methods for all the synthetic input matrices, almost  
 467 uniformly and sometimes by orders of magnitude.
- 468 • For input matrices where the spectral tail decays slowly (PolyDecaySlow,  
 469 LowRankLowNoise, LowRankMedNoise, LowRankHiNoise), the newly proposed  
 470 method (2.10) has identical behavior to the Upadhyay method (6.1).
- 471 • For input matrices whose spectral tail decays more quickly (ExpDecaySlow,  
 472 ExpDecayMed, ExpDecayFast, PolyDecayMed, PolyDecayFast), the proposed  
 473 method improves massively over Upadhyay (6.1).
- 474 • The new method (2.10) shows its strength over the two-sketch method (6.2)  
 475 when the storage budget is small. It also yields superior performance in  
 476 Schatten  $\infty$ -norm. These differences are most evident for matrices with slow  
 477 spectral decay.

478 In summary, the proposed method (2.10) enjoys the advantages of the Upad-  
 479 hyay (6.1) method and our previous approach (6.2), with no evident disadvantages.

### 480 6.7. Comparison of Reconstruction Formulas: Real Data Examples.

481 Our last set of experiments is designed to show that our sketching and reconstruction  
 482 pipeline is effective for real data.

483 Figures 7 and 8 contains the results of the following experiment. For each of  
 484 the three rank- $r$  reconstruction methods, we display the relative error (6.3) as a  
 485 function of storage. We use sparse dimension reduction maps, which is justified by  
 486 the experiments in subsection 6.4.

We plot the oracle error (subsection 6.2.2) attained by each method. Since the  
 oracle error is not achievable in practice, we also chart the performance of each method  
 at an *a priori* selection of parameters. For the proposed method (2.10), we use  
 the natural parameter choice (5.6) that follows from our theoretical analysis. The  
 Upadhyay sketch takes the same form as ours but lacks a comparable theory, so we  
 instantiate his method with the same parameters (5.6) we used in our sketch. Last,  
 for the two-sketch method (6.2), we assume that the input matrix  $\mathbf{A} \in \mathbb{F}^{m \times n}$  is tall  
 ( $m \geq n$ ), and we use the theoretically motivated parameter values

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

487 This choice adapts the arguments in [46, Sec. 4.5.2] to use the current definition of  
 488 the storage budget  $T$ .

489 As with the synthetic examples, the proposed method (2.10) dominates the com-  
 490 peting methods for all the examples we considered. This is true when we compare  
 491 oracle errors or when we compare the errors using *a priori* parameter choices. The  
 492 benefits of the new method are least pronounced for the matrix MinTemp, whose spec-

493 trum has medium polynomial decay. The benefits of the new method are quite clear  
 494 for the matrix `StreamVel`, which has an exponentially decaying spectrum. The ad-  
 495 vantages are even more striking for the two matrices `MaxCut` and `PhaseRetrieval`,  
 496 which are effectively rank deficient.

497 In summary, we believe that the numerical work here supports the use of our new  
 498 method (2.10). The Upadhyay (6.1) method cannot achieve a small relative error (6.3),  
 499 even with a large amount of storage. The two-sketch method (6.2) can achieve small  
 500 relative error, but it often requires more storage to achieve this goal—especially at  
 501 the *a priori* parameter choices.

502 **6.8. Example: Flow-Field Reconstruction.** Finally, we elaborate on using  
 503 sketching to compress the DNS data matrix `StreamVel`. We compute the best rank-  
 504 10 approximation of the matrix via (2.10) using storage  $T/(m+n) = 48$  and the  
 505 parameter choices (5.6). For this example, we can use plots of the flow field to make  
 506 visual comparisons.

507 Figure 9 illustrates the leading left singular vectors of the streamwise velocity  
 508 field `StreamVel`, as computed from the sketch and the full matrix. We see that  
 509 the approximate left singular vectors closely match the actual left singular vectors,  
 510 although some small errors appear, especially at the inlet (on the left-hand side of the  
 511 images). See subsection SM3.4 for additional numerics.

512 If we normalize `StreamVel` so that its largest singular value equals one, then  
 513 the best rank-10 approximation of `StreamVel` has absolute  $S_\infty$  error  $2.223 \cdot 10^{-2}$ .  
 514 Meanwhile, the computed rank-10 approximation has absolute  $S_\infty$  error  $2.226 \cdot 10^{-2}$ .  
 515 (The  $S_\infty$  relative error (6.3) is  $1.3 \cdot 10^{-3}$ .) We can easily improve these numbers by  
 516 computing a higher-rank approximation and/or increasing the storage budget.

517 We learn that the sketched matrix supports an excellent rank-10 reconstruction,  
 518 even though it only uses 5.8MB of storage in double precision. For comparison,  
 519 the full matrix requires 409.7MB of storage. The compression rate is  $70.6\times$ . This  
 520 demonstration suggests that it is indeed possible to automatically compress the output  
 521 of the DNS simulation using sketching.

522 **7. Conclusions.** This paper exhibits a sketching method and a new reconstruc-  
 523 tion algorithm for low-rank approximation of matrices that are presented as a sequence  
 524 of linear updates (section 2). We have described how to implement the method using  
 525 SSRFTs or sparse dimension reduction methods (section 3), and we have argued that  
 526 the performance of the method is insensitive to the choice of dimension reduction map  
 527 (subsection 6.4). In addition, a detailed theoretical analysis (section 5) prescribes how  
 528 to select parameter values for the sketch *a priori*, and we have shown that these pa-  
 529 rameter values yield good performance across a range of examples (subsection 6.5).  
 530 Finally, we have demonstrated that the new reconstruction method dominates existing  
 531 techniques for both synthetic matrices (subsection 6.6) and real data (subsection 6.7).

532 A potential application of these techniques is for on-the-fly-compression of data  
 533 from large-scale simulations. Our work with DNS data indicates that we can achieve  
 534 significant data reduction. A key advantage of our new approach over (6.2) is that it  
 535 extends to higher-dimensional (i.e., tensor) data. This generalization should allow for  
 536 higher compression rates, and we plan to explore this idea in a future work.

537 **Acknowledgments.** The authors wish to thank Beverley McKeon and Sean  
 538 Symon for providing DNS simulation data and visualization software. William North  
 539 contributed the weather data.

540

## REFERENCES

- 541 [1] D. ACHLIOPTAS, *Database-friendly random projections: Johnson–Lindenstrauss with binary*  
542 *coins*, J. Comput. System Sci., 66 (2003), pp. 671–687.
- 543 [2] N. AILON AND B. CHAZELLE, *The fast Johnson–Lindenstrauss transform and approximate*  
544 *nearest neighbors*, SIAM J. Comput., 39 (2009), pp. 302–322, [https://doi.org/10.1137/](https://doi.org/10.1137/060673096)  
545 [060673096](http://dx.doi.org/10.1137/060673096), <http://dx.doi.org/10.1137/060673096>.
- 546 [3] W. AUSTIN, G. BALLARD, AND T. G. KOLDA, *Parallel tensor compression for large-scale scienti-*  
547 *fic data*, in 2016 IEEE Intl. Symp. Parallel and Distributed Processing, 2016, pp. 912–922.
- 548 [4] A. H. BAKER, H. XU, J. M. DENNIS, M. N. LEVY, D. NYCHKA, S. A. MICKELSON, J. EDWARDS,  
549 M. VERTENSTEIN, AND A. WEGENER, *A methodology for evaluating the impact of data*  
550 *compression on climate simulation data*, in Proc. 23rd ACM Intl. Symp. High-Performance  
551 Parallel and Distributed Computing, 2014, pp. 203–214.
- 552 [5] R. BAURLE, *Modeling of high speed reacting flows: Established practices and future challenges*,  
553 in 42nd AIAA Aerospace Sciences Meeting and Exhibit, 2004, p. 267.
- 554 [6] A. BEJAN, *Convection heat transfer*, John Wiley & Sons, 2013.
- 555 [7] J. BOURGAIN, S. DIRKSEN, AND J. NELSON, *Toward a unified theory of sparse dimensionality*  
556 *reduction in Euclidean space*, Geom. Funct. Anal., 25 (2015), pp. 1009–1088, [https://doi.](https://doi.org/10.1007/s00039-015-0332-9)  
557 [org/10.1007/s00039-015-0332-9](http://dx.doi.org/10.1007/s00039-015-0332-9), <http://dx.doi.org/10.1007/s00039-015-0332-9>.
- 558 [8] C. BOUTSIDIS AND A. GITTENS, *Improved matrix algorithms via the subsampled randomized*  
559 *Hadamard transform*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1301–1340, [https://doi.](https://doi.org/10.1137/120874540)  
560 [org/10.1137/120874540](http://dx.doi.org/10.1137/120874540), <http://dx.doi.org/10.1137/120874540>.
- 561 [9] C. BOUTSIDIS, D. WOODRUFF, AND P. ZHONG, *Optimal principal component analysis in dis-*  
562 *tributed and streaming models*. Available at <http://arXiv.org/abs/1504.06729>, July 2016.
- 563 [10] C. BOUTSIDIS, D. WOODRUFF, AND P. ZHONG, *Optimal principal component analysis in dis-*  
564 *tributed and streaming models*, in Proc. 48th ACM Symp. Theory of Computing (STOC  
565 2016), Cambridge, MA, 2016.
- 566 [11] J. CALHOUN, F. CAPPELLO, L. N. OLSON, M. SNIR, AND W. D. GROPP, *Exploring the feasi-*  
567 *bility of lossy compression for PDE simulations*, The International Journal of High Perfor-
- 568 *mance Computing Applications*, (2018), p. 1094342018762036, [https://doi.org/10.1177/](https://doi.org/10.1177/1094342018762036)  
569 [1094342018762036](https://doi.org/10.1177/1094342018762036).
- 570 [12] S. CASTRUCCIO AND M. G. GENTON, *Compressing an ensemble with statistical models: An*  
571 *algorithm for global 3D spatio-temporal temperature*, Technometrics, 58 (2016), pp. 319–  
572 328.
- 573 [13] K. L. CLARKSON AND D. P. WOODRUFF, *Numerical linear algebra in the streaming model*, in  
574 Proc. 41st ACM Symp. Theory of Computing (STOC), Bethesda, 2009.
- 575 [14] K. L. CLARKSON AND D. P. WOODRUFF, *Low rank approximation and regression in input spar-*  
576 *sity time*, in Proc. 45th ACM Symp. Theory of Computing (STOC), ACM, New York, 2013,  
577 pp. 81–90, <https://doi.org/10.1145/2488608.2488620>, [http://dx.doi.org/10.1145/2488608.](http://dx.doi.org/10.1145/2488608.2488620)  
578 [2488620](http://dx.doi.org/10.1145/2488620).
- 579 [15] M. COHEN, *Nearly tight oblivious subspace embeddings by trace inequalities*, in Proc. 27th  
580 ACM-SIAM Symp. Discrete Algorithms (SODA), Arlington, Jan. 2016, pp. 278–287.
- 581 [16] M. B. COHEN, S. ELDER, C. MUSCO, C. MUSCO, AND M. PERSU, *Dimensionality reduction*  
582 *for k-means clustering and low rank approximation*, in Proc. 47th ACM Symp. Theory of  
583 Computing (STOC), ACM, 2015, pp. 163–172.
- 584 [17] J. B. DRAKE, *Climate modeling for scientists and engineers*, SIAM, 2014.
- 585 [18] D. FELDMAN, M. VOLKOV, AND D. RUS, *Dimensionality reduction of massive sparse datasets*  
586 *using coresets*, in Adv. Neural Information Processing Systems 29 (NIPS), 2016.
- 587 [19] A. FRIEZE, R. KANNAN, AND S. VEMPALA, *Fast Monte-Carlo algorithms for finding low-rank*  
588 *approximations*, J. Assoc. Comput. Mach., 51 (2004), pp. 1025–1041, [https://doi.org/10.](https://doi.org/10.1145/1039488.1039494)  
589 [1145/1039488.1039494](http://dx.doi.org/10.1145/1039488.1039494), <http://dx.doi.org/10.1145/1039488.1039494>.
- 590 [20] E. GARNIER, N. ADAMS, AND P. SAGAUT, *Large eddy simulation for compressible flows*, Springer  
591 Science & Business Media, 2009.
- 592 [21] M. GHASHAMI, E. LIBERTY, J. M. PHILLIPS, AND D. P. WOODRUFF, *Frequent directions:*  
593 *simple and deterministic matrix sketching*, SIAM J. Comput., 45 (2016), pp. 1762–  
594 1792, <https://doi.org/10.1137/15M1009718>, [https://doi-org.clsproxy.library.caltech.edu/](https://doi-org.clsproxy.library.caltech.edu/10.1137/15M1009718)  
595 [10.1137/15M1009718](https://doi-org.clsproxy.library.caltech.edu/10.1137/15M1009718).
- 596 [22] M. X. GOEMANS AND D. WILLIAMSON, *Improved approximation algorithms for maximum cut*  
597 *and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42  
598 (1995), pp. 1115–1145.
- 599 [23] J. GUINNESS AND D. HAMMERLING, *Compression and conditional emulation of climate model*  
600 *output*, J. Amer. Stat. Assoc., (2017).

- 601 [24] N. HALKO, P.-G. MARTINSSON, Y. SHKOLNISKY, AND M. TYGERT, *An algorithm for the prin-*  
 602 *icipal component analysis of large data sets*, SIAM J. Sci. Comput., 33 (2011), pp. 2580–  
 603 2594, <https://doi.org/10.1137/100804139>, [https://doi-org.clsproxy.library.caltech.edu/10.](https://doi-org.clsproxy.library.caltech.edu/10.1137/100804139)  
 604 [1137/100804139](https://doi.org/10.1137/100804139).
- 605 [25] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: prob-*  
 606 *abilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53  
 607 (2011), pp. 217–288.
- 608 [26] N. J. HIGHAM, *Matrix nearness problems and applications*, in Applications of matrix theory  
 609 (Bradford, 1988), Oxford Univ. Press, New York, 1989, pp. 1–27.
- 610 [27] R. HORSTMAYER, R. Y. CHEN, X. OU, B. AMES, J. A. TROPP, AND C. YANG, *Solving ptychog-*  
 611 *raphy with a convex relaxation*, New J. Physics, 17 (2015), p. 053044.
- 612 [28] P. INDYK AND R. MOTWANI, *Approximate nearest neighbors: Towards removing the curse*  
 613 *of dimensionality*, in Proc. 30th ACM Symp. Theory of Computing (STOC), STOC '98,  
 614 New York, NY, USA, 1998, ACM, pp. 604–613, <https://doi.org/10.1145/276698.276876>,  
 615 <http://doi.acm.org/10.1145/276698.276876>.
- 616 [29] W. B. JOHNSON AND J. LINDENSTRAUSS, *Extensions of Lipschitz mapping into Hilbert space*,  
 617 Contemp. Math., 26 (1984), pp. 189–206.
- 618 [30] Y. LI, H. L. NGUYEN, AND D. P. WOODRUFF, *Turnstile streaming algorithms might as well*  
 619 *be linear sketches*, in Proc. 46th ACM Symp. Theory of Computing (STOC), ACM, New  
 620 York, 2014, pp. 174–183.
- 621 [31] E. LIBERTY, *Accelerated dense random projections*, PhD thesis, Yale Univ., New Haven, 2009.
- 622 [32] M. W. MAHONEY, *Randomized algorithms for matrices and data*, Found. Trends Mach. Learn-  
 623 ing, 3 (2011), pp. 123–224.
- 624 [33] M. R. MALIK, B. J. ISAAC, A. COUSSEMENT, P. J. SMITH, AND A. PARENTE, *Principal compo-*  
 625 *nent analysis coupled with nonlinear regression for chemistry reduction*, Combustion and  
 626 Flame, 187 (2018), pp. 30–41.
- 627 [34] P.-G. MARTINSSON, V. ROKHLIN, AND M. TYGERT, *A randomized algorithm for the de-*  
 628 *composition of matrices*, Appl. Comput. Harmon. Anal., 30 (2011), pp. 47–68, <https://doi.org/10.1016/j.acha.2010.02.003>,  
 629 <http://dx.doi.org/10.1016/j.acha.2010.02.003>.
- 630 [35] X. MENG AND M. W. MAHONEY, *Low-distortion subspace embeddings in input-sparsity time and*  
 631 *applications to robust linear regression*, in Proc. 45th ACM Symp. Theory of Computing  
 632 (STOC), ACM, New York, 2013, pp. 91–100, <https://doi.org/10.1145/2488608.2488621>,  
 633 <http://dx.doi.org/10.1145/2488608.2488621>.
- 634 [36] F. R. MENTER, M. KUNTZ, AND R. LANGTRY, *Ten years of industrial experience with the sst*  
 635 *turbulence model*, Turbulence, heat and mass transfer, 4 (2003), pp. 625–632.
- 636 [37] J. NELSON AND H. L. NGUYEN, *OSNAP: faster numerical linear algebra algorithms via sparser*  
 637 *subspace embeddings*, in 2013 IEEE 54th Symp. Foundations of Computer Science (FOCS),  
 638 IEEE Computer Soc., Los Alamitos, CA, 2013, pp. 117–126, [https://doi.org/10.1109/](https://doi.org/10.1109/FOCS.2013.21)  
 639 [FOCS.2013.21](https://doi.org/10.1109/FOCS.2013.21), <http://dx.doi.org/10.1109/FOCS.2013.21>.
- 640 [38] J. NELSON AND H. L. NGUYEN, *Lower bounds for oblivious subspace embeddings*, in Automata,  
 641 languages, and programming. Part I, vol. 8572 of Lecture Notes in Comput. Sci., Springer,  
 642 Heidelberg, 2014, pp. 883–894, [https://doi.org/10.1007/978-3-662-43948-7\\_73](https://doi.org/10.1007/978-3-662-43948-7_73), [http://dx.](http://dx.doi.org/10.1007/978-3-662-43948-7_73)  
 643 [doi.org/10.1007/978-3-662-43948-7\\_73](http://dx.doi.org/10.1007/978-3-662-43948-7_73).
- 644 [39] C. H. PAPADIMITRIOU, P. RAGHAVAN, H. TAMAKI, AND S. VEMPALA, *Latent semantic in-*  
 645 *dexing: a probabilistic analysis*, J. Comput. System Sci., 61 (2000), pp. 217–235, <https://doi.org/10.1006/jcss.2000.1711>,  
 646 <http://dx.doi.org/10.1006/jcss.2000.1711>. Special issue  
 647 on the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of  
 648 Database Systems (Seattle, WA, 1998).
- 649 [40] S. PATANKAR, *Numerical heat transfer and fluid flow*, CRC press, 1980.
- 650 [41] V. ROKHLIN, A. SZLAM, AND M. TYGERT, *A randomized algorithm for principal component*  
 651 *analysis*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1100–1124, [https://doi.org/10.1137/](https://doi.org/10.1137/080736417)  
 652 [080736417](https://doi.org/10.1137/080736417), <https://doi-org.clsproxy.library.caltech.edu/10.1137/080736417>.
- 653 [42] P. SAGAUT, *Large eddy simulation for incompressible flows: an introduction*, Springer Science  
 654 & Business Media, 2006.
- 655 [43] S. W. SON, Z. CHEN, W. HENDRIX, A. AGRAWAL, W.-K. LIAO, AND A. CHOUDHARY, *Data com-*  
 656 *pression for the exascale computing era-survey*, Supercomputing frontiers and innovations,  
 657 1 (2014), pp. 76–88.
- 658 [44] J. A. TROPP, *Improved analysis of the subsampled randomized Hadamard transform*, Adv.  
 659 Adapt. Data Anal., 3 (2011), pp. 115–126, <https://doi.org/10.1142/S1793536911000787>,  
 660 <http://dx.doi.org/10.1142/S1793536911000787>.
- 661 [45] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Fixed-rank approximation of a*  
 662 *positive-semidefinite matrix from streaming data*, in Adv. Neural Information Processing

- 663 Systems 30 (NIPS), Long Beach, Dec. 2017.
- 664 [46] J. A. TROP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Practical sketching algorithms for*  
665 *low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485.
- 666 [47] J. A. TROP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Randomized single-view algorithms*  
667 *for low-rank matrix approximation*, ACM Report 2017-01, Caltech, Pasadena, Jan. 2017.  
668 Available at <http://arXiv.org/abs/1609.00048>, v1.
- 669 [48] J. UPADHYAY, *Fast and space-optimal low-rank factorization in the streaming model with ap-*  
670 *plication in differential privacy*. Available at <http://arXiv.org/abs/1604.01429>, Apr. 2016.
- 671 [49] J. WOODRING, S. MNISZEWSKI, C. BRISLAWN, D. DEMARLE, AND J. AHRENS, *Revisiting wavelet*  
672 *compression for large-scale climate data using JPEG 2000 and ensuring data precision*, in  
673 2011 IEEE Symp. Large Data Analysis and Visualization (LDAV), 2011, pp. 31–38.
- 674 [50] D. P. WOODRUFF, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor.  
675 Comput. Sci., 10 (2014), pp. iv+157.
- 676 [51] F. WOOLFE, E. LIBERTY, V. ROKHLIN, AND M. TYGERT, *A fast randomized algorithm for the*  
677 *approximation of matrices*, Appl. Comput. Harmon. Anal., 25 (2008), pp. 335–366.
- 678 [52] A. YURTSEVER, M. UDELL, J. A. TROP, AND V. CEVHER, *Sketchy decisions: Convex low-rank*  
679 *matrix optimization with optimal storage*, in 2017 Intl. Conf. Artificial Intelligence and  
680 Statistics (AISTATS), 2017.
- 681 [53] G. ZHOU, A. CICHOCKI, AND S. XIE, *Decomposition of big tensors with low multilinear rank*,  
682 arXiv preprint arXiv:1412.1885, (2014).

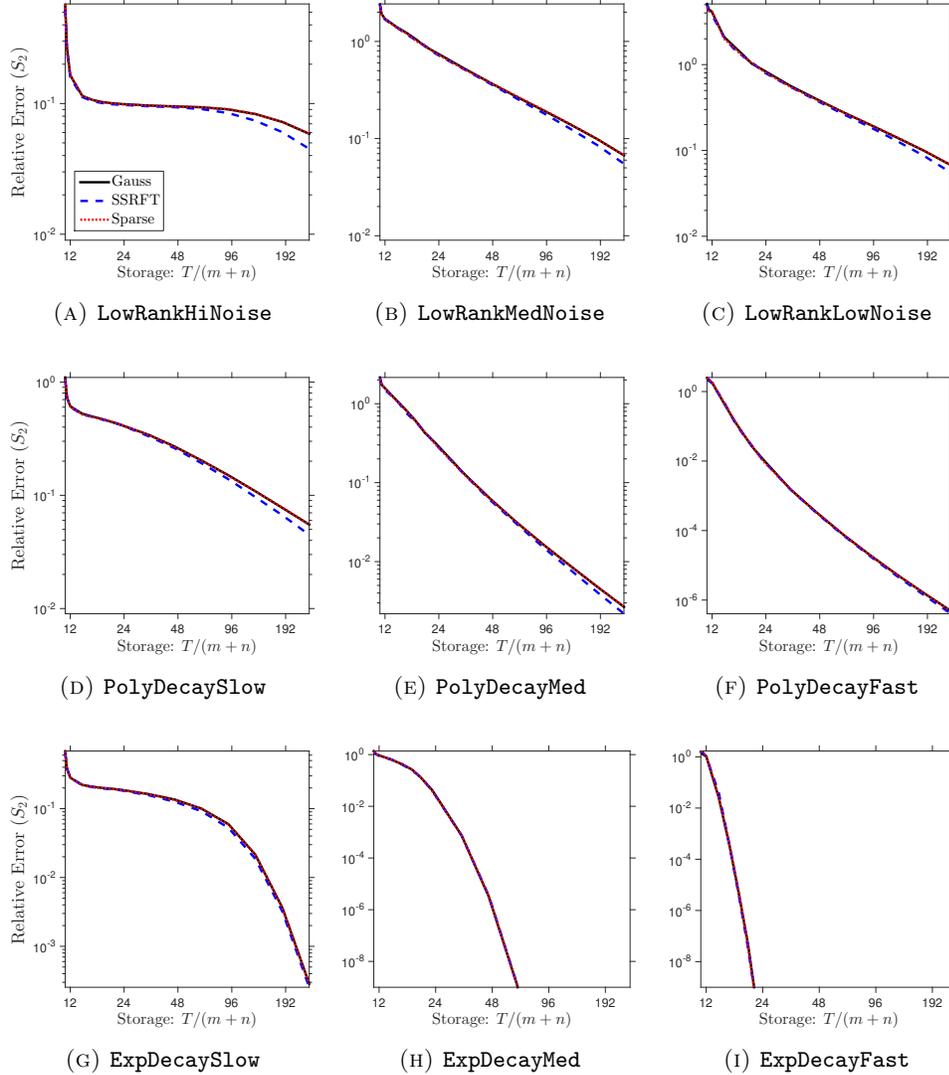


FIG. 2: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection 6.4 for details.

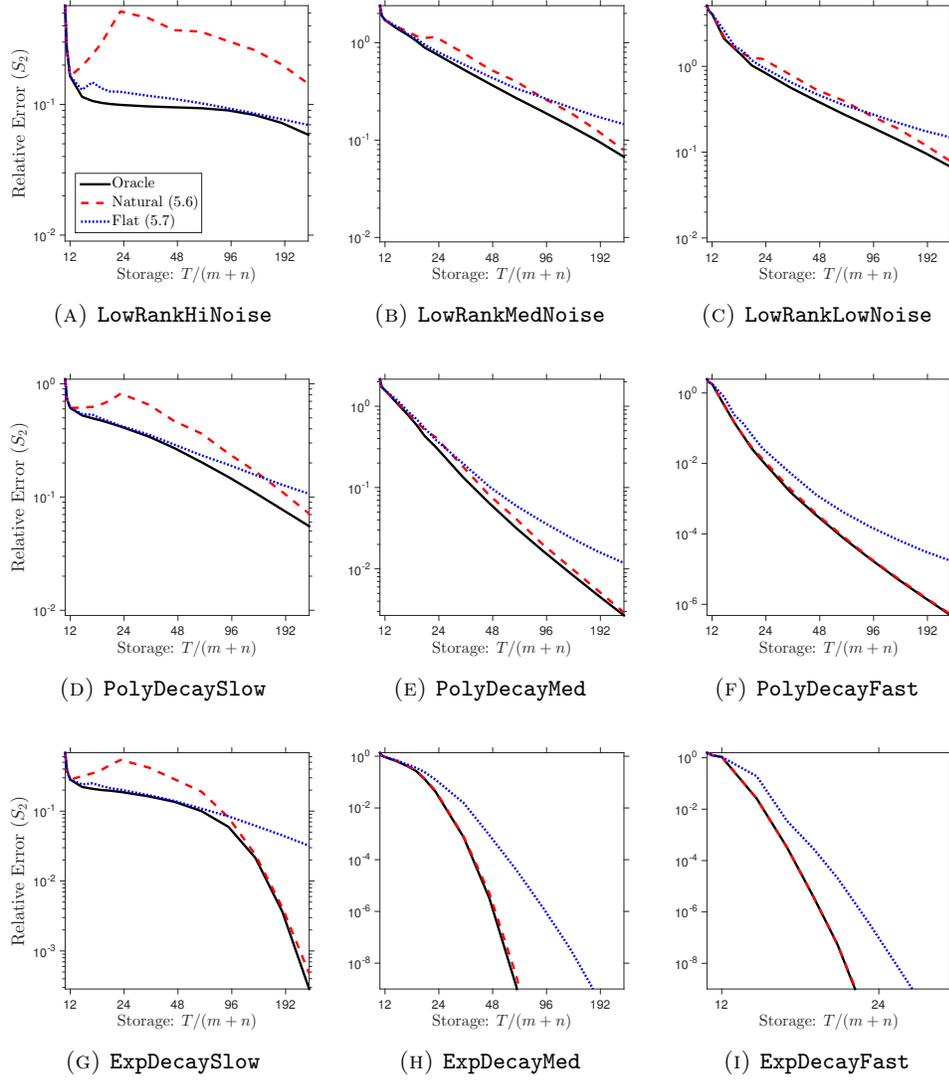


FIG. 3: **Relative error for proposed method with *a priori* parameters.** (Gaussian maps, effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

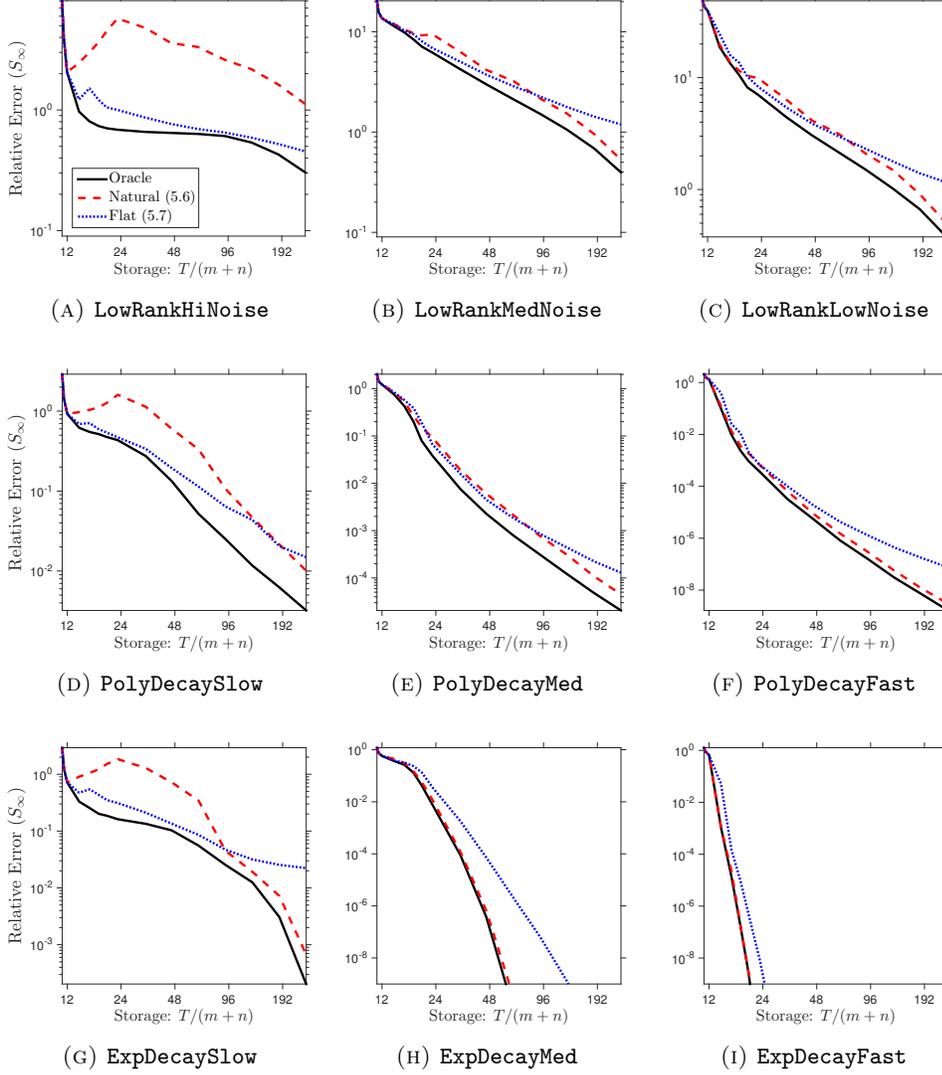


FIG. 4: Relative error for proposed method with *a priori* parameters. (Gaussian maps, effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

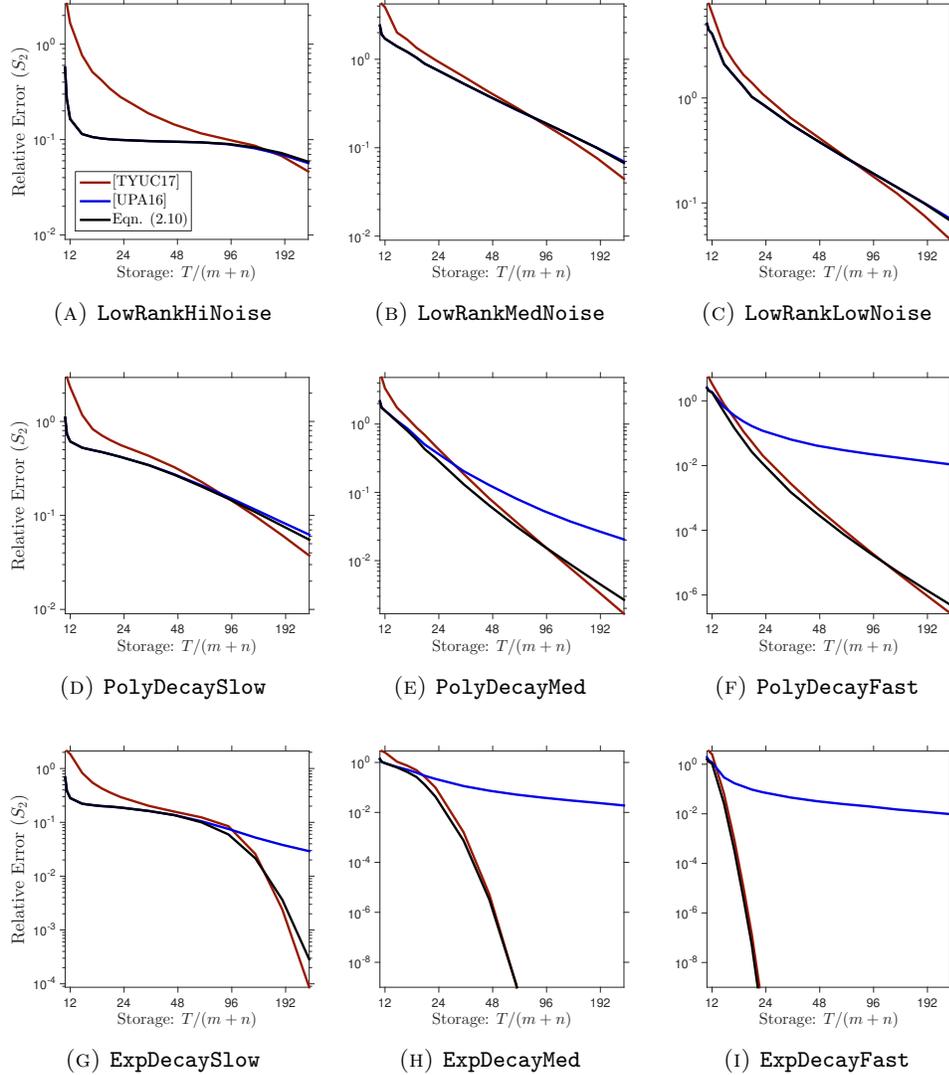


FIG. 5: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See subsection 6.2.2 for details.

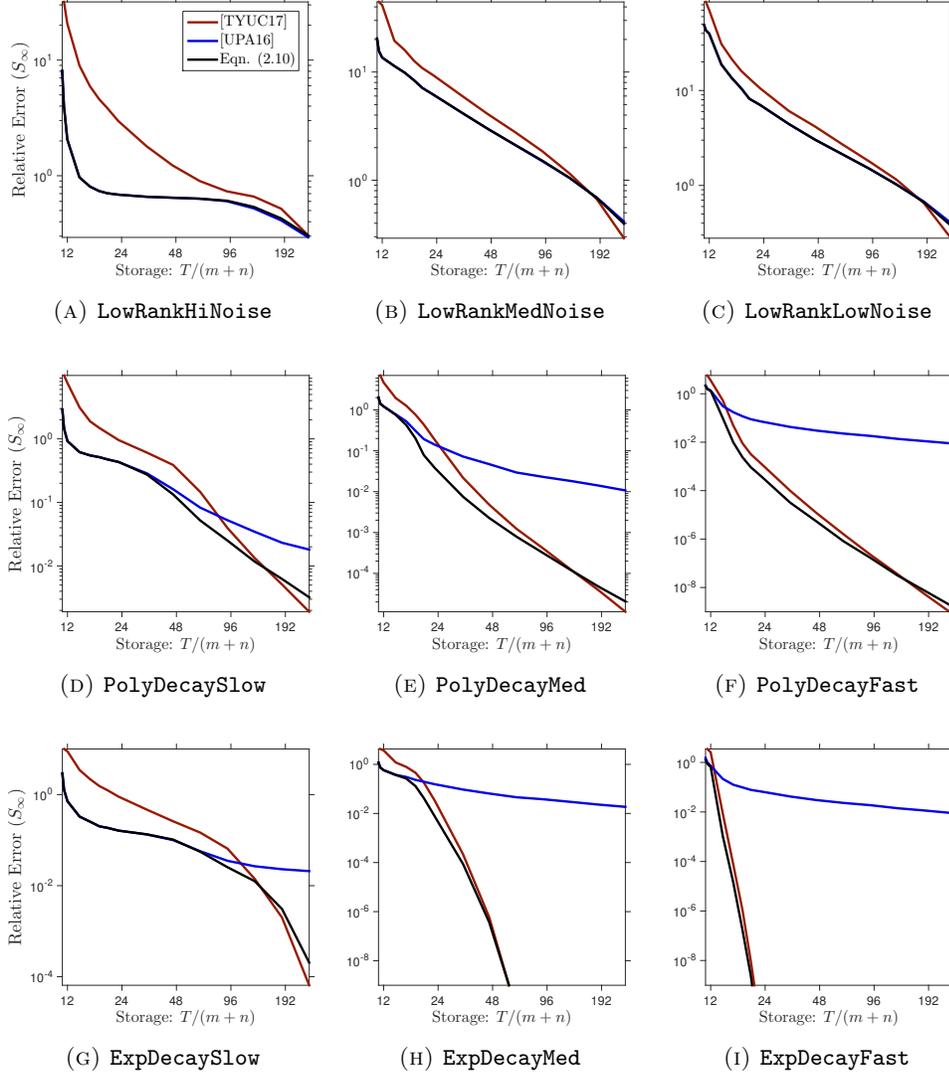


FIG. 6: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See subsection 6.2.2 for details.

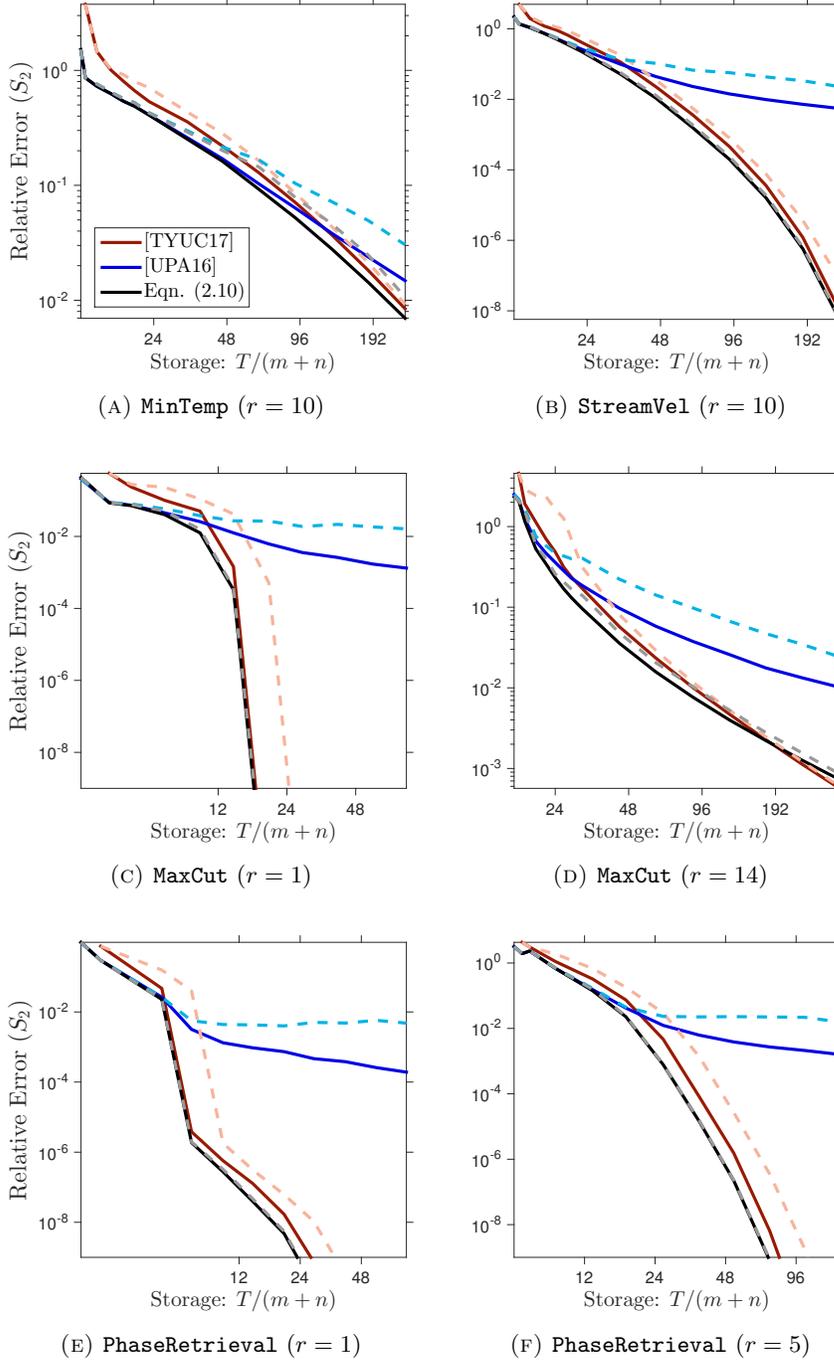


FIG. 7: **Comparison of reconstruction formulas: Real data examples.** (Sparse maps, Schatten 2-norm.) We compare the relative error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. **Solid lines** are oracle errors; **dashed lines** are errors with “natural” parameter choices. See subsection 6.7 for details.

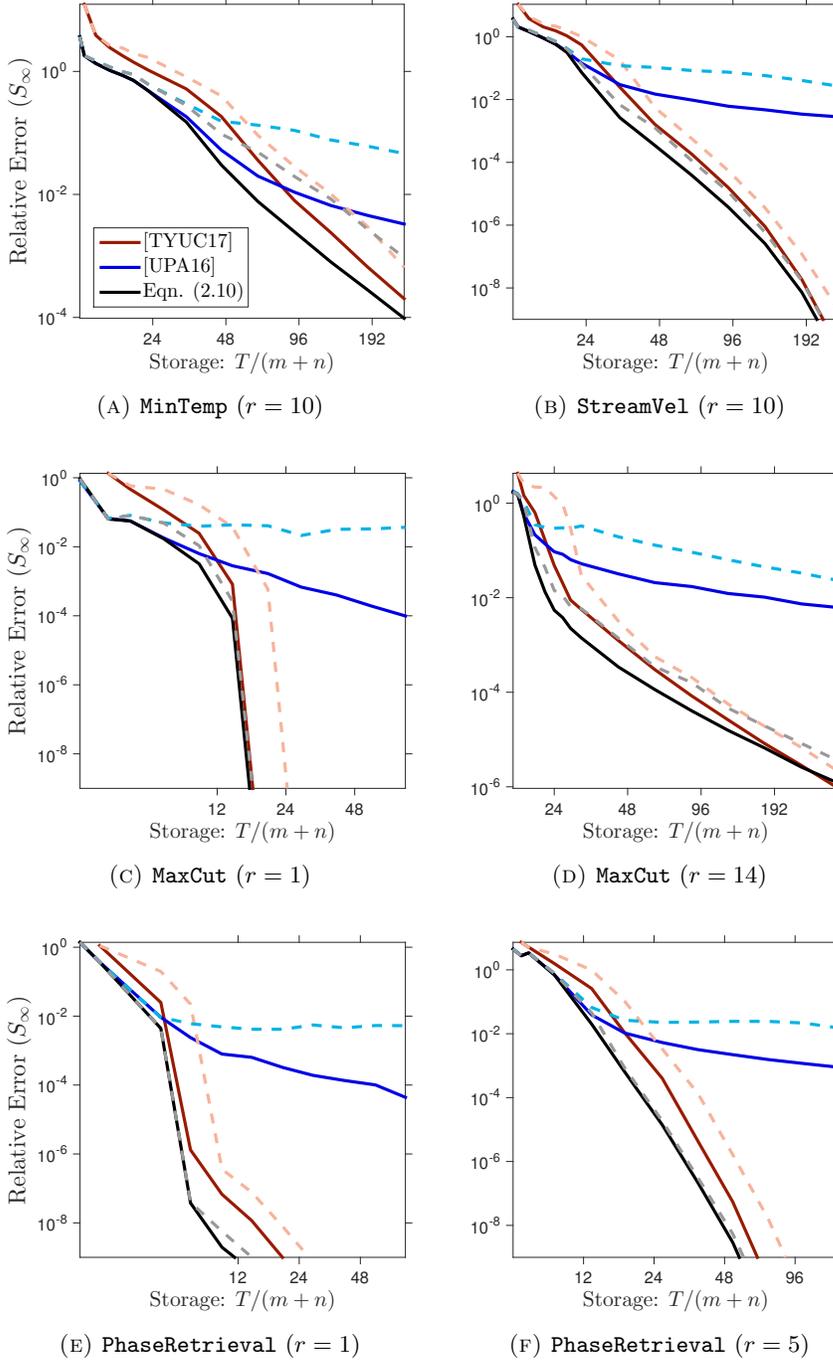


FIG. 8: **Comparison of reconstruction formulas: Real data examples.** (Sparse maps, Schatten  $\infty$ -norm.) We compare the relative error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. **Solid lines** are oracle errors; **dashed lines** are errors with “natural” parameter choices. See subsection 6.7 for details.

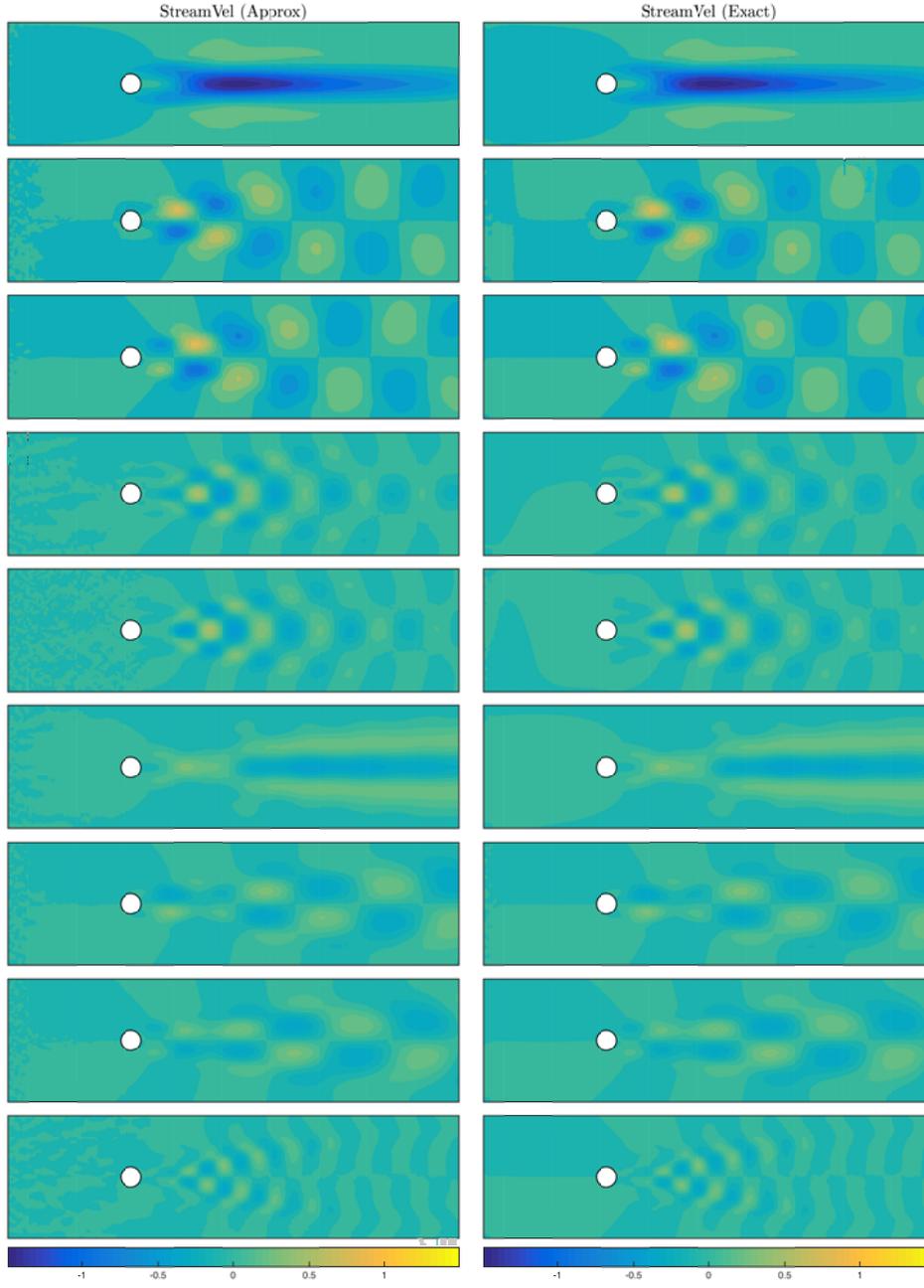


FIG. 9: **Left singular vectors of StreamVel.** (Sparse maps, approximation rank  $r = 10$ , storage budget  $T = 48(m + n)$ .) The columns of the matrix **StreamVel** describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show the first nine computed left singular vectors of the matrix. The **left-hand side** is computed from the sketch, while the **right-hand side** is computed from the exact flow field. The heatmap indicates the magnitude of the fluctuation. See [subsection 6.8](#) for details.

1 **SUPPLEMENTARY MATERIALS: MORE PRACTICAL SKETCHING**  
2 **ALGORITHMS FOR LOW-RANK MATRIX APPROXIMATION\***

3 JOEL A. TROPP<sup>†</sup>, ALP YURTSEVER<sup>‡</sup>, MADELEINE UDELL<sup>§</sup>, AND VOLKAN CEVHER<sup>‡</sup>

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

7 **SM1.1. Facts about Random Matrices.** First, let us state a useful formula  
8 that allows us to compute some expectations involving a Gaussian random matrix.  
9 This identity is drawn from [[SM1](#), Prop. A.1 and A.6]. See also [[SM2](#), Fact A.1].

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

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

10 *The number  $\alpha$  is given by [\(5.1\)](#).*

11 **SM1.2. Results from Randomized Linear Algebra.** Our argument also  
12 depends on the analysis of randomized low-rank approximation developed in [[SM1](#),  
13 Sec. 10].

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

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

14 *The number  $\alpha$  is given by [\(5.1\)](#).*

15 This result follows immediately from the proof of [[SM1](#), Thm. 10.5] using [Fact SM1.1](#)  
16 to handle both the real and complex case simultaneously. See also [[SM3](#), Sec. 8.2].

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

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

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

---

\*First draft: 22 March 2017. First release: 16 July 2018.

**Funding:** JAT was supported in part by ONR Awards N00014-11-1002, N00014-17-1-214, N00014-17-1-2146, and the Gordon & Betty Moore Foundation. MU was supported in part by DARPA Award FA8750-17-2-0101. VC has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under the grant agreement number 725594 (time-data) and the Swiss National Science Foundation (SNSF) under the grant number 200021\_178865.

<sup>†</sup>California Institute of Technology, Pasadena, CA ([jtropp@cms.caltech.edu](mailto:jtropp@cms.caltech.edu)).

<sup>‡</sup>École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland ([alp.yurtsever@epfl.ch](mailto:alp.yurtsever@epfl.ch), [volkan.cevher@epfl.ch](mailto:volkan.cevher@epfl.ch)).

<sup>§</sup>Cornell University, Ithaca, NY ([udell@cornell.edu](mailto:udell@cornell.edu)).

21 The columns of  $P_\perp$  and  $Q_\perp$  are orthonormal. Next, introduce the submatrices

$$22 \quad (\text{SM1.1}) \quad \begin{aligned} \Phi_1 &= \Phi Q \in \mathbb{F}^{s \times k} & \text{and} & & \Phi_2 &= \Phi Q_\perp \in \mathbb{F}^{s \times (m-k)}; \\ \Psi_1^* &= P^* \Psi^* \in \mathbb{F}^{k \times s} & \text{and} & & \Psi_2^* &= P_\perp^* \Psi^* \in \mathbb{F}^{(n-k) \times s}. \end{aligned}$$

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

24 **LEMMA SM1.3** (Decomposition of the Core Matrix Approximation). *Assume*  
25 *that the matrices  $\Phi_1$  and  $\Psi_1$  have full column rank. Then*

$$26 \quad \begin{aligned} W - Q^* A P &= \Phi_1^\dagger \Phi_2 (Q_\perp^* A P) + (Q^* A P_\perp) \Psi_2^* (\Psi_1^\dagger)^* \\ 27 &+ \Phi_1^\dagger \Phi_2 (Q_\perp^* A P_\perp) \Psi_2^* (\Psi_1^\dagger)^*. \end{aligned}$$

*Proof.* Adding and subtracting terms, we write the core sketch  $Z$  as

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

Using (SM1.1), we identify the matrices  $\Phi_1$  and  $\Psi_1$ . Then left-multiply by  $\Phi_1^\dagger$  and right-multiply by  $(\Psi_1^\dagger)^*$  to arrive at

$$W = \Phi_1^\dagger Z (\Psi_1^\dagger)^* = \Phi_1^\dagger \Phi (A - Q Q^* A P P^*) \Psi^* (\Psi_1^\dagger)^* + Q^* A P.$$

29 We have identified the core matrix  $W$ , defined in (2.8). Move the term  $Q^* A P$  to the  
30 left-hand side to isolate the approximation error.

To continue, notice that

$$\Phi_1^\dagger \Phi = \Phi_1^\dagger \Phi Q Q^* + \Phi_1^\dagger \Phi Q_\perp Q_\perp^* = Q^* + \Phi_1^\dagger \Phi_2 Q_\perp^*.$$

Likewise,

$$\Psi^* (\Psi_1^\dagger)^* = P P^* \Psi^* (\Psi_1^\dagger)^* + P_\perp P_\perp^* \Psi^* (\Psi_1^\dagger)^* = P + P_\perp \Psi_2^* (\Psi_1^\dagger)^*.$$

Combine the last three displays to arrive at

$$W - Q^* A P = (Q^* + \Phi_1^\dagger \Phi_2 Q_\perp^*) (A - Q Q^* A P P^*) (P + P_\perp \Psi_2^* (\Psi_1^\dagger)^*).$$

31 Expand the expression and use the orthogonality relations  $Q^* Q = \mathbf{I}$  and  $Q_\perp^* Q = \mathbf{0}$   
32 and  $P^* P = \mathbf{I}$  and  $P^* P_\perp = \mathbf{0}$  to arrive at the desired representation.  $\square$

33 **SM1.4. Probabilistic Analysis of the Core Matrix.** Next, we make distri-  
34 butional assumptions on the dimension reduction maps  $\Phi$  and  $\Psi$ . We can then study  
35 the probabilistic behavior of the error  $W - Q^* A P$ .

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

$$\mathbb{E}_{\Phi, \Psi} [W - Q^* A P] = \mathbf{0}.$$

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

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

36 When  $s \geq 2k + \alpha$ , the last term is always nonpositive.

37 *Proof.* Since  $\Phi$  is standard normal, the orthogonal submatrices  $\Phi_1$  and  $\Phi_2$  are  
 38 statistically independent standard normal matrices because of the marginal property  
 39 of the normal distribution. Likewise,  $\Psi_1$  and  $\Psi_2$  are statistically independent stan-  
 40 dard normal matrices. Provided that  $s \geq k$ , both matrices have full column rank with  
 41 probability one.

42 To establish the first point, notice that

$$43 \quad \mathbb{E}_{\Phi, \Psi} [\mathbf{W} - \mathbf{Q}^* \mathbf{A} \mathbf{P}] = \mathbb{E}_{\Phi_1} \mathbb{E}_{\Phi_2} [\Phi_1^\dagger \Phi_2 (\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P})] + \mathbb{E}_{\Psi_1} \mathbb{E}_{\Psi_2} [(\mathbf{Q}^* \mathbf{A} \mathbf{P}_\perp) \Psi_2^* (\Psi_1^\dagger)^*]  
 44 \quad + \mathbb{E}_{\Phi_1} \mathbb{E}_{\Phi_2} [\Phi_1^\dagger \Phi_2 (\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp) \Psi_2^* (\Psi_1^\dagger)^*].$$

46 We have used the decomposition of the approximation error from [Lemma SM1.3](#). Then  
 47 we invoke independence to write the expectations as iterated expectations. Since  $\Phi_2$   
 48 and  $\Psi_2$  have mean zero, this formula makes it clear that the approximation error has  
 49 mean zero.

50 To study the fluctuations, apply the independence and zero-mean property of  $\Phi_2$   
 51 and  $\Psi_2$  to decompose

$$52 \quad \mathbb{E}_{\Phi, \Psi} \|\mathbf{W} - \mathbf{Q}^* \mathbf{A} \mathbf{P}\|_2^2 = \mathbb{E}_{\Phi} \|\Phi_1^\dagger \Phi_2 (\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P})\|_2^2 + \mathbb{E}_{\Psi} \|(\mathbf{Q}^* \mathbf{A} \mathbf{P}_\perp) \Psi_2^* (\Psi_1^\dagger)^*\|_2^2  
 53 \quad + \mathbb{E}_{\Phi} \mathbb{E}_{\Psi} \|\Phi_1^\dagger \Phi_2 (\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp) \Psi_2^* (\Psi_1^\dagger)^*\|_2^2.$$

55 Continuing, we invoke [Fact SM1.1](#) four times to see that

$$56 \quad \mathbb{E}_{\Phi, \Psi} \|\mathbf{W} - \mathbf{Q}^* \mathbf{A} \mathbf{P}\|_2^2  
 57 \quad = \frac{k}{s - k - \alpha} \cdot \left[ \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}\|_2^2 + \|\mathbf{Q}^* \mathbf{A} \mathbf{P}_\perp\|_2^2 + \frac{k}{s - k - \alpha} \cdot \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp\|_2^2 \right].$$

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

$$61 \quad \mathbb{E} \|\mathbf{W} - \mathbf{Q}^* \mathbf{A} \mathbf{P}\|_2^2 = \frac{k}{s - k - \alpha} \cdot \left[ \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}\|_2^2 + \|\mathbf{Q}^* \mathbf{A} \mathbf{P}_\perp\|_2^2 + \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp\|_2^2  
 62 \quad + \frac{2k + \alpha - s}{s - k - \alpha} \cdot \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp\|_2^2 \right].$$

64 Use the Pythagorean Theorem to combine the terms on the first line.  $\square$

65 **SM1.5. Probabilistic Analysis of the Compression Error.** Next, we es-  
 66 tablish a bound for the expected error in the compression of the matrix  $\mathbf{A}$  onto the  
 67 range of the matrices  $\mathbf{Q}$  and  $\mathbf{P}$ , computed in [\(2.7\)](#). This result is similar in spirit to  
 68 the analysis in [\[SM1\]](#), so we pass lightly over the details.

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

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

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

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^* = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^* \\ \mathbf{V}_2^* \end{bmatrix} \quad \text{where } \Sigma_1 \in \mathbb{F}^{\varrho \times \varrho}.$$

69 Define the matrices

$$\begin{aligned}
70 \quad & \Upsilon_1 := \Upsilon U_1 \in \mathbb{F}^{s \times \varrho} \quad \text{and} \quad \Upsilon_2 := \Upsilon U_2 \in \mathbb{F}^{s \times (m-\varrho)}; \\
71 \quad & \Omega_1^* := V_1^* \Omega^* \in \mathbb{F}^{\varrho \times s} \quad \text{and} \quad \Omega_2^* := V_2^* \Omega^* \in \mathbb{F}^{(n-\varrho) \times s}; \\
72 \quad & P_1 := V_1^* P \in \mathbb{F}^{\varrho \times k} \quad \text{and} \quad P_2 := V_2^* P \in \mathbb{F}^{(n-\varrho) \times k}.
\end{aligned}$$

74 With this notation, we proceed to the proof.

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

$$\|A - QQ^* APP^*\|_2^2 = \|A(\mathbf{I} - PP^*)\|_2^2 + \|(\mathbf{I} - QQ^*)APP^*\|_2^2.$$

75 Use the SVD to decompose the matrix  $A$  in the first term, and apply the Pythagorean  
76 Theorem again:

$$\begin{aligned}
77 \quad & \|A - QQ^* APP^*\|_2^2 = \|(U_2 \Sigma_2 V_2^*)(\mathbf{I} - PP^*)\|_2^2 \\
78 \quad & \quad + \|(U_1 \Sigma_1 V_1^*)(\mathbf{I} - PP^*)\|_2^2 + \|(\mathbf{I} - QQ^*)AP\|_2^2.
\end{aligned}$$

The result [SM3, Prop. 9.2] implies that the second term satisfies

$$\|(U_1 \Sigma_1 V_1^*)(\mathbf{I} - PP^*)\|_2^2 \leq \|\Upsilon_1^\dagger \Upsilon_2 \Sigma_2\|_2^2.$$

We can obtain a bound for the third term using the formula [SM1, p. 270, disp. 1].  
After a short computation, this result yields

$$\begin{aligned}
\|(\mathbf{I} - QQ^*)AP\|_2^2 & \leq \|\Sigma_2 P_2\|_2^2 + \|\Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger P_1\|_2^2 \\
& \leq \|(U_2 \Sigma_2 V_2^*)P\|_2^2 + \|\Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger\|_2^2.
\end{aligned}$$

We can remove  $P_1$  because its spectral norm is bounded by one, being a submatrix  
of an orthonormal matrix. Combine the last three displays to obtain

$$\|A - QQ^* APP^*\|_2^2 \leq \|U_2 \Sigma_2 V_2^*\|_2^2 + \|\Upsilon_1^\dagger \Upsilon_2 \Sigma_2\|_2^2 + \|\Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger\|_2^2.$$

81 We have used the Pythagorean Theorem again.

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

$$\begin{aligned}
\mathbb{E} \|A - QQ^* APP^*\|_2^2 & \leq \|\Sigma_2\|_2^2 + \mathbb{E} \|\Upsilon_1^\dagger \Upsilon_2 \Sigma_2\|_2^2 + \mathbb{E} \|\Sigma_2 \Omega_2^* (\Omega_1^*)^\dagger\|_2^2 \\
& = \|\Sigma_2\|_2^2 + \frac{2\varrho}{k - \varrho - \alpha} \cdot \|\Sigma_2\|_2^2.
\end{aligned}$$

82 Finally, note that  $\|\Sigma_2\|_2^2 = \tau_{\varrho+1}^2(A)$ . □

**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  $\hat{A}$ , we see that

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

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

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

83 We treat the two terms sequentially.

To continue, invoke the expression [Lemma SM1.4](#) for the expected error in the core matrix  $\mathbf{W}$ :

$$\begin{aligned} \mathbb{E}_{\Phi, \Psi} \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 &\leq \left(1 + \frac{k}{s - k - \alpha}\right) \cdot \|\mathbf{A} - \mathbf{Q}\mathbf{Q}^* \mathbf{A} \mathbf{P}\mathbf{P}^*\|_2^2 \\ &\quad + \frac{k(2k + \alpha - s)}{(s - k - \alpha)^2} \cdot \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp\|_2^2. \end{aligned}$$

84 Now, take the expectation with respect to  $\Upsilon$  and  $\Omega$  to arrive at

85 (SM1.2)

$$\begin{aligned} \mathbb{E} \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 &\leq \left(1 + \frac{k}{s - k - \alpha}\right) \cdot \left(1 + \frac{2\varrho}{k - \varrho - \alpha}\right) \cdot \tau_{\varrho+1}^2(\mathbf{A}) \\ &\quad + \frac{k(2k + \alpha - s)}{(s - k - \alpha)^2} \cdot \mathbb{E} \|\mathbf{Q}_\perp^* \mathbf{A} \mathbf{P}_\perp\|_2^2. \end{aligned}$$

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

90 **SM2. Code & Pseudocode.** This supplement contains pseudocode for the  
 91 sketching and low-rank reconstruction algorithms described in this paper. In many  
 92 places, we use the same mathematical notation as the rest of the paper. We also rely  
 93 on MATLAB R2018A commands, which appear in typewriter font. The electronic  
 94 materials include a MATLAB implementation of these methods.

- 95 • [Algorithm SM3.1](#) contains the constructor for the SKETCH object. It draws  
 96 random test matrices and initializes the sketch for the zero input matrix. This  
 97 code implements (2.2)–(2.4).
- 98 • [Algorithm SM3.2](#) implements a general rank-one linear update (2.5) to the  
 99 input matrix contained in the sketch.
- 100 • [Algorithm SM3.3](#) implements the basic low-rank reconstruction formula (2.9).  
 101 It returns the approximation in factored form.
- 102 • [Algorithm SM3.4](#) implements the rank- $r$  reconstruction formula (2.10). It  
 103 returns the approximation in factored form.
- 104 • [Algorithm SM3.5](#) is the template for the dimension reduction (DIMREDUX)  
 105 class for input matrices over the field  $\mathbb{F}$ . It outlines the methods that a  
 106 DIMREDUX needs to implement.
- 107 • [Algorithm SM3.6](#) defines the Gaussian dimension reduction (GAUSS) class,  
 108 which is a subclass of DIMREDUX. It describes the constructor and the left  
 109 and right action of this dimension reduction map. See [subsection 3.1](#) for the  
 110 explanation.
- 111 • [Algorithm SM3.7](#) defines the SSRFT dimension reduction (SSRFT) class,  
 112 which is a subclass of DIMREDUX. It describes the constructor and the left  
 113 and right action of this dimension reduction map. See [subsection 3.2](#) for the  
 114 explanation.
- 115 • [Algorithm SM3.8](#) defines the sparse dimension reduction (SPARSE) class,  
 116 which is a subclass of DIMREDUX. It describes the constructor and the left  
 117 and right action of this dimension reduction map. See [subsection 3.3](#) for the  
 118 explanation.

119 **SM3. Supplemental Numerical Results.** This section summarizes the addi-  
120 tional numerical results that are presented in this supplement. The MATLAB code in  
121 the electronic materials can reproduce these experiments.

122 **SM3.1. Insensitivity to the Dimension Reduction Map.** We undertook  
123 a more comprehensive set of experiments to demonstrate that our reconstruction  
124 formula (2.10) is insensitive to the choice of dimension reduction map at the oracle  
125 parameters. See subsection 6.4 for details.

126 Figures SM1 to SM5 contain the results for matrices with effective rank  $R = 5$ ,  
127  $R = 10$ , and  $R = 20$  with relative error measured in Schatten 2-norm and Schatten  
128  $\infty$ -norm.

129 **SM3.2. Achieving the Oracle Performance.** We also performed experi-  
130 ments to see how closely the theoretical parameter choices allow us to approach the  
131 oracle performance of our reconstruction formula (2.10). See subsection 6.5 for details.

132 Figures SM6 to SM9 contain the results for matrices with effective rank  $R = 5$   
133 and  $R = 20$  with relative error measured in Schatten 2-norm and Schatten  $\infty$ -norm.

134 **SM3.3. Algorithm Comparisons for Synthetic Instances.** We compared  
135 all three of the reconstruction formulas (2.10), (6.1), and (6.2) at the oracle parameters  
136 for a wide range of synthetic problem instances. See subsection 6.6 for details.

137 Figures SM10 to SM13 contain the results for matrices with effective rank  $R = 5$   
138 and  $R = 20$  with relative error measured in Schatten 2-norm and Schatten  $\infty$ -norm.

139 **SM3.4. Flow-Field Reconstruction.** Figure SM14 illustrates the streamwise  
140 velocity field `StreamVel` and its rank-10 approximation via (2.10) using storage budget  
141  $T/(m+n) = 48$  and the parameter choices (5.6). We see that the approximation  
142 captures the large-scale features of the flow, although there are small errors visible,  
143 especially at the inlet (on the left-hand side of the images).

144

## REFERENCES

- 145 [SM1] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: prob-*  
146 *abilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53  
147 (2011), pp. 217–288.  
148 [SM2] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Practical sketching algorithms for*  
149 *low-rank matrix approximation*. Submitted, Jan. 2017.  
150 [SM3] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Randomized single-view algorithms*  
151 *for low-rank matrix approximation*, ACM Report 2017-01, Caltech, Pasadena, Jan. 2017.  
152 Available at <http://arXiv.org/abs/1609.00048>, v1.

---

**Algorithm SM3.1** *Sketch for Low-Rank Approximation.* Implements (2.2)–(2.4).

---

**Input:** Input matrix dimensions  $m \times n$ ; sketch size parameters  $k \leq s \leq \min\{m, n\}$

**Output:** Draw dimension reduction maps (2.2); sketch (2.3) and (2.4) of  $\mathbf{A} = \mathbf{0}$

```

1 class SKETCH
2   local variables  $\Upsilon, \Omega, \Phi, \Psi$  (DIMREDUX)
3   local variables  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$  (matrices)
4   function SKETCH( $m, n, k, s$ ; DR)      ▷ Constructor; DR is a DIMREDUX
5      $\Upsilon \leftarrow \text{DR}(k, m)$           ▷ Draw new dimension reduction maps
6      $\Omega \leftarrow \text{DR}(k, n)$ 
7      $\Phi \leftarrow \text{DR}(s, m)$ 
8      $\Psi \leftarrow \text{DR}(s, n)$ 
9      $\mathbf{X} \leftarrow \text{zeros}(k, n)$           ▷ Sketch of zero matrix
10     $\mathbf{Y} \leftarrow \text{zeros}(m, k)$ 
11     $\mathbf{Z} \leftarrow \text{zeros}(s, s)$ 

```

---



---

**Algorithm SM3.2** *Linear Update to Sketch.* Implements (2.5).

---

**Input:** Innovation  $\mathbf{H} \in \mathbb{F}^{m \times n}$ ; scalars  $\theta, \tau \in \mathbb{F}$

**Output:** Modifies sketch to reflect linear update  $\mathbf{A} \leftarrow \theta\mathbf{A} + \tau\mathbf{H}$

```

1 function SKETCH.LINEARUPDATE( $\mathbf{H}; \theta, \tau$ )
2    $\mathbf{X} \leftarrow \theta\mathbf{X} + \tau\Upsilon\mathbf{H}$ 
3    $\mathbf{Y} \leftarrow \theta\mathbf{Y} + \tau\mathbf{H}\Omega^*$ 
4    $\mathbf{Z} \leftarrow \theta\mathbf{Z} + \tau\Phi\mathbf{H}\Psi^*$ 

```

---



---

**Algorithm SM3.3** *Low-Rank Approximation.* Implements (2.9).

---

**Output:** Rank- $k$  approximation of sketched matrix in form  $\hat{\mathbf{A}} = \mathbf{Q}\mathbf{W}\mathbf{P}^*$  with orthonormal  $\mathbf{Q} \in \mathbb{F}^{m \times k}$  and  $\mathbf{P} \in \mathbb{F}^{n \times k}$

```

1 function SKETCH.LOWRANKAPPROX()
2    $(\mathbf{Q}, \sim) \leftarrow \text{qr}(\mathbf{Y}, 0)$ 
3    $(\mathbf{P}, \sim) \leftarrow \text{qr}(\mathbf{X}^*, 0)$ 
4    $\mathbf{W} \leftarrow ((\Phi\mathbf{Q}) \setminus \mathbf{Z}) / ((\Psi\mathbf{P})^*)$           ▷ Least-squares via QR or SVD
5   return  $(\mathbf{Q}, \mathbf{W}, \mathbf{P})$ 

```

---

---

**Algorithm SM3.4** *Fixed-Rank Approximation.* Implements (2.10).

---

**Input:** Rank  $r$  of approximation

**Output:** Rank- $r$  approximation of sketched matrix in form  $\hat{\mathbf{A}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$  with orthonormal  $\mathbf{U} \in \mathbb{F}^{n \times r}$  and  $\mathbf{V} \in \mathbb{F}^{m \times r}$  and nonnegative diagonal  $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$

```

1 function SKETCH.FIXEDRANKAPPROX( $r$ )
2   ( $\mathbf{Q}, \mathbf{W}, \mathbf{P}$ )  $\leftarrow$  SKETCH.LOWRANKAPPROX()
3   ( $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ )  $\leftarrow$  svds( $\mathbf{W}, r$ )            $\triangleright$  Truncate full SVD to rank  $r$ 
4    $\mathbf{U} \leftarrow \mathbf{Q}\mathbf{U}$                                 $\triangleright$  Consolidate unitary factors
5    $\mathbf{V} \leftarrow \mathbf{P}\mathbf{V}$ 
6   return ( $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ )

```

---



---

**Algorithm SM3.5** *Dimension Reduction Map Class.*

---

```

1 class DIMREDUX ( $\mathbb{F}$ )                                $\triangleright$  Dimension reduction map over field  $\mathbb{F}$ 
2   function DIMREDUX( $k, n$ )                           $\triangleright$  Construct map  $\mathbf{\Xi} : \mathbb{F}^n \rightarrow \mathbb{F}^k$ 
3   function DIMREDUX.MTIMES(DRmap,  $\mathbf{M}$ )               $\triangleright$  Left action of map
4   function DIMREDUX.MTIMES( $\mathbf{M}, \text{DRmap}^*$ )           $\triangleright$  Right action of adjoint
5   return (DIMREDUX.MTIMES(DRmap,  $\mathbf{M}^*$ ))*           $\triangleright$  Default behavior

```

---



---

**Algorithm SM3.6** *Gaussian Dimension Reduction Map.* (subsection 3.1)

---

```

1 class GAUSS (DIMREDUX)                              $\triangleright$  Subclass of DIMREDUX
2   local variable  $\mathbf{\Xi}$  (dense matrix)
3   function RANDN( $k, n; \mathbb{F}$ )                           $\triangleright$  Gaussian matrix over field  $\mathbb{F}$ 
4     if  $\mathbb{F} = \mathbb{R}$  then return randn( $k, n$ )
5     if  $\mathbb{F} = \mathbb{C}$  then return randn( $k, n$ ) + 1i * randn( $k, n$ )
6   function GAUSS( $k, n$ )                                $\triangleright$  Constructor
7      $\mathbf{\Xi} \leftarrow$  RANDN( $k, n; \mathbb{F}$ )                    $\triangleright$  Gaussian over  $\mathbb{F}$ 
8   function GAUSS.MTIMES(DRmap,  $\mathbf{M}$ )
9     return mtimes( $\mathbf{\Xi}, \mathbf{M}$ )

```

---

---

**Algorithm SM3.7** *SSRFT Dimension Reduction Map.* (subsection 3.2)

---

```

1 class SSRFT (DIMREDUX)                                ▷ Subclass of DIMREDUX
2 local variables coords, permj, εj for j = 1, 2

3 function SSRFT(k, n)                                  ▷ Constructor
4   coords ← randperm(n, k)
5   permj ← randperm(n) for j = 1, 2
6   εj ← sign(RANDN(n, 1;  $\mathbb{F}$ )) for j = 1, 2

7 function SSRFT.MTIMES(DRmap,  $M$ )
8   if  $\mathbb{F} = \mathbb{R}$  then
9      $M \leftarrow \text{dct}(\text{diag}(\varepsilon_1)M(\text{perm}_1, :))$ 
10     $M \leftarrow \text{dct}(\text{diag}(\varepsilon_2)M(\text{perm}_2, :))$ 
11  if  $\mathbb{F} = \mathbb{C}$  then
12     $M \leftarrow \text{dft}(\text{diag}(\varepsilon_1)M(\text{perm}_1, :))$ 
13     $M \leftarrow \text{dft}(\text{diag}(\varepsilon_2)M(\text{perm}_2, :))$ 
14  return  $M(\text{coords}, :)$ 

```

---



---

**Algorithm SM3.8** *Sparse Dimension Reduction Map.* (subsection 3.3)

---

```

1 class SPARSE (DIMREDUX)                                ▷ Subclass of DIMREDUX
2 local variable  $\Xi$  (sparse matrix)

3 function SPARSE(k, n)                                  ▷ Constructor
4    $\zeta \leftarrow \min\{k, \lfloor 2 \log(1 + n) \rfloor\}$           ▷ Sparsity of each column
5   for j = 1, ..., n do
6      $\Xi(\text{randperm}(k, \zeta), j) \leftarrow \text{sign}(\text{RANDN}(\zeta, 1; \mathbb{F}))$ 

7 function SPARSE.MTIMES(DRmap,  $M$ )
8   return mtimes( $\Xi$ ,  $M$ )

```

---

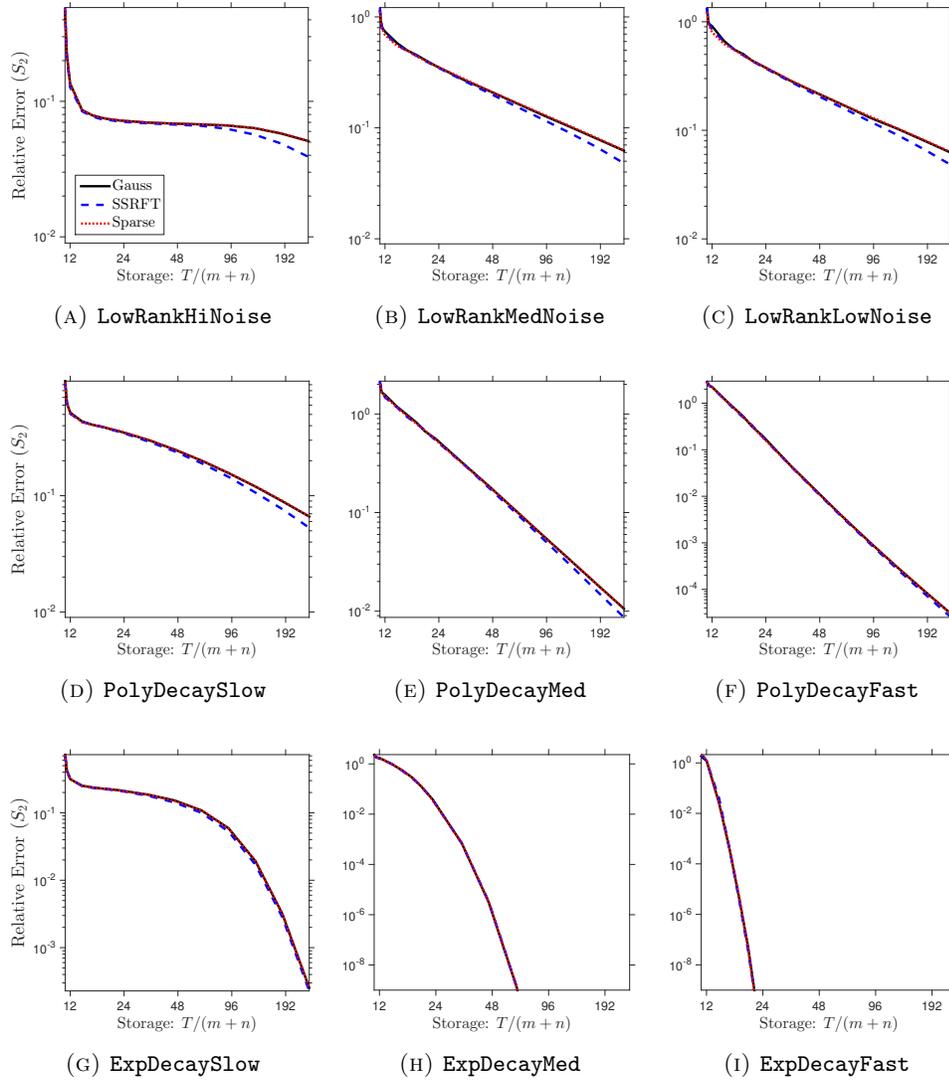


FIG. SM1: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See [subsection 6.4](#) for details.

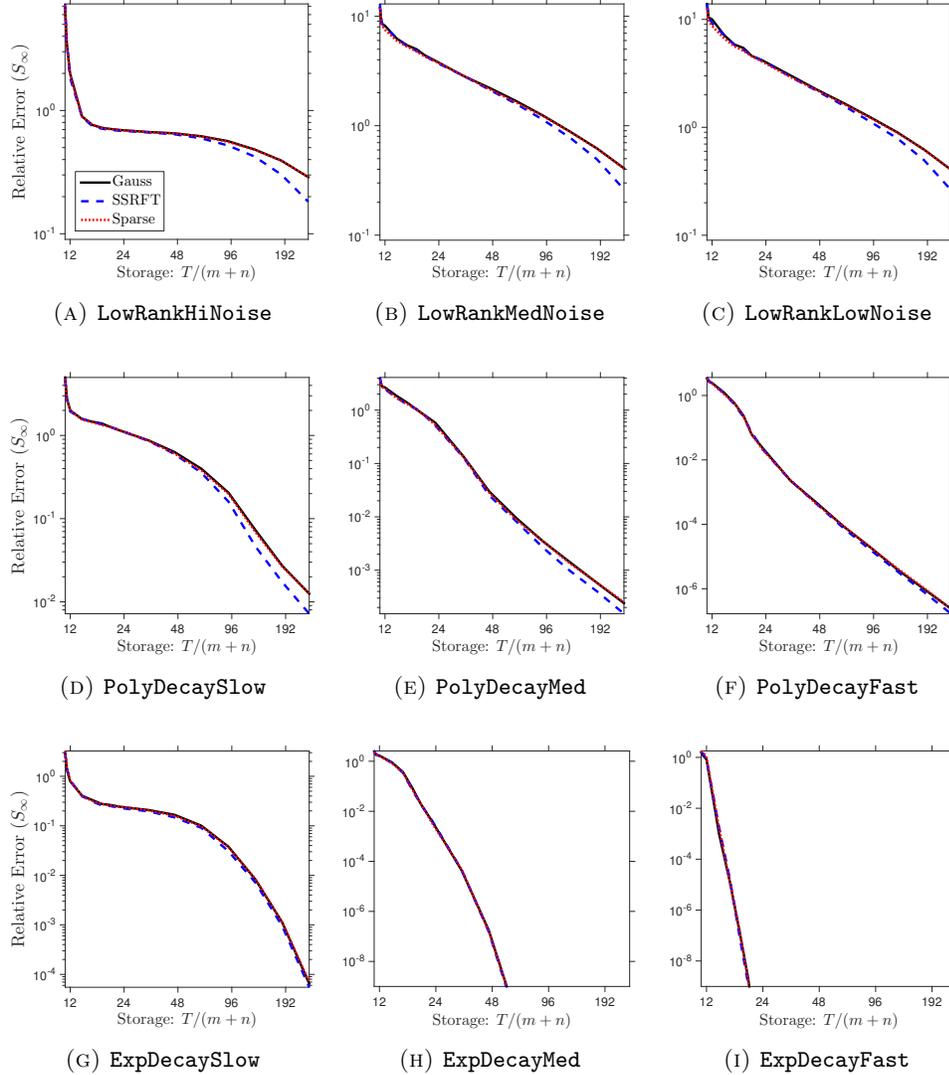


FIG. SM2: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection 6.4 for details.

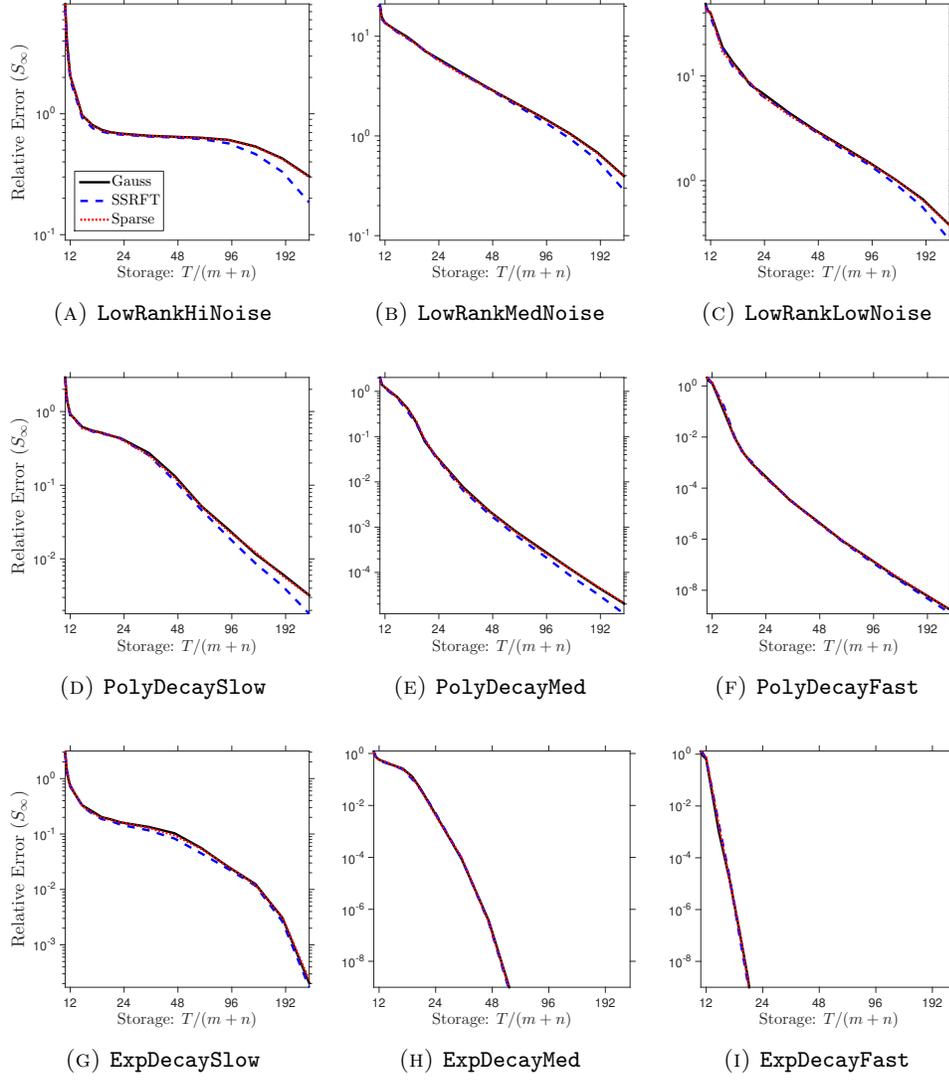


FIG. SM3: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 10$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection 6.4 for details.

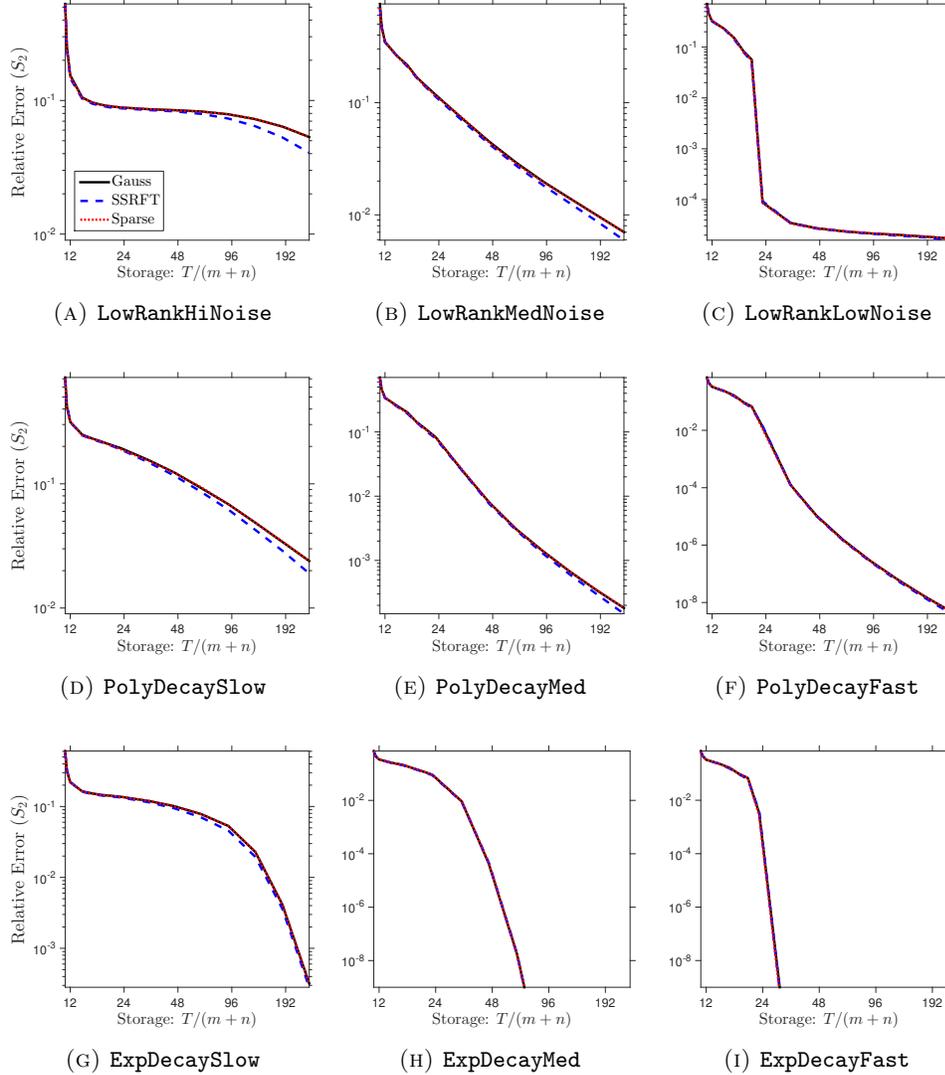


FIG. SM4: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection 6.4 for details.

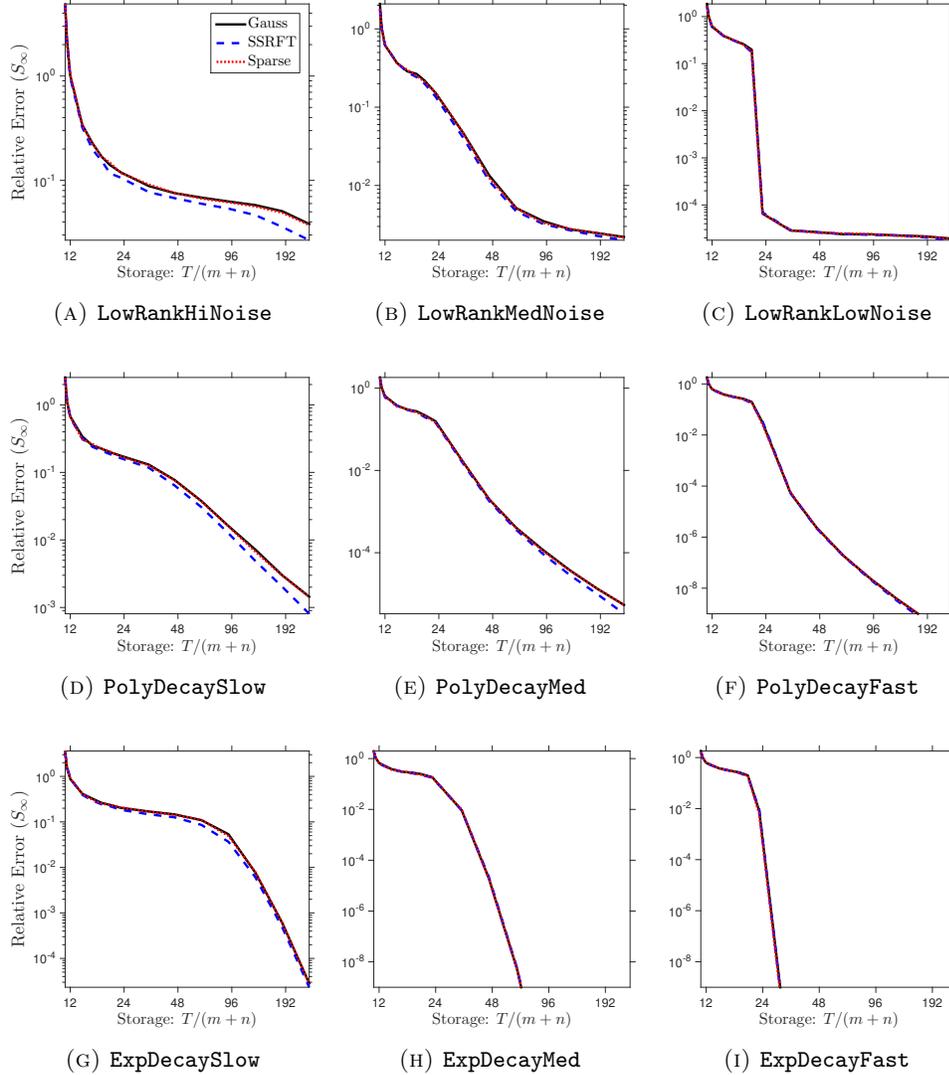


FIG. SM5: **Insensitivity of proposed method to the dimension reduction map.** (Effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) implemented with Gaussian, SSRFT, or sparse dimension reduction maps. See subsection 6.4 for details.

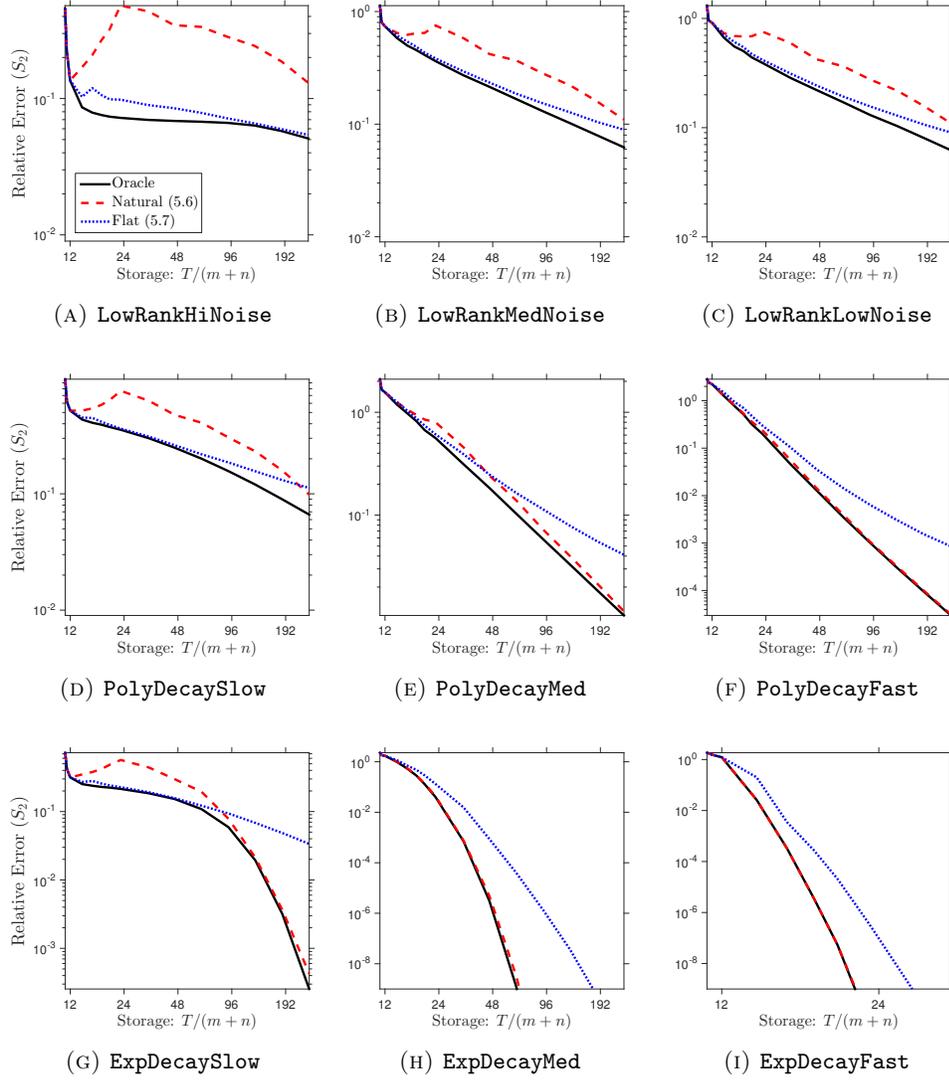


FIG. SM6: **Relative error for proposed method with *a priori* parameters.** (Gaussian maps, effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

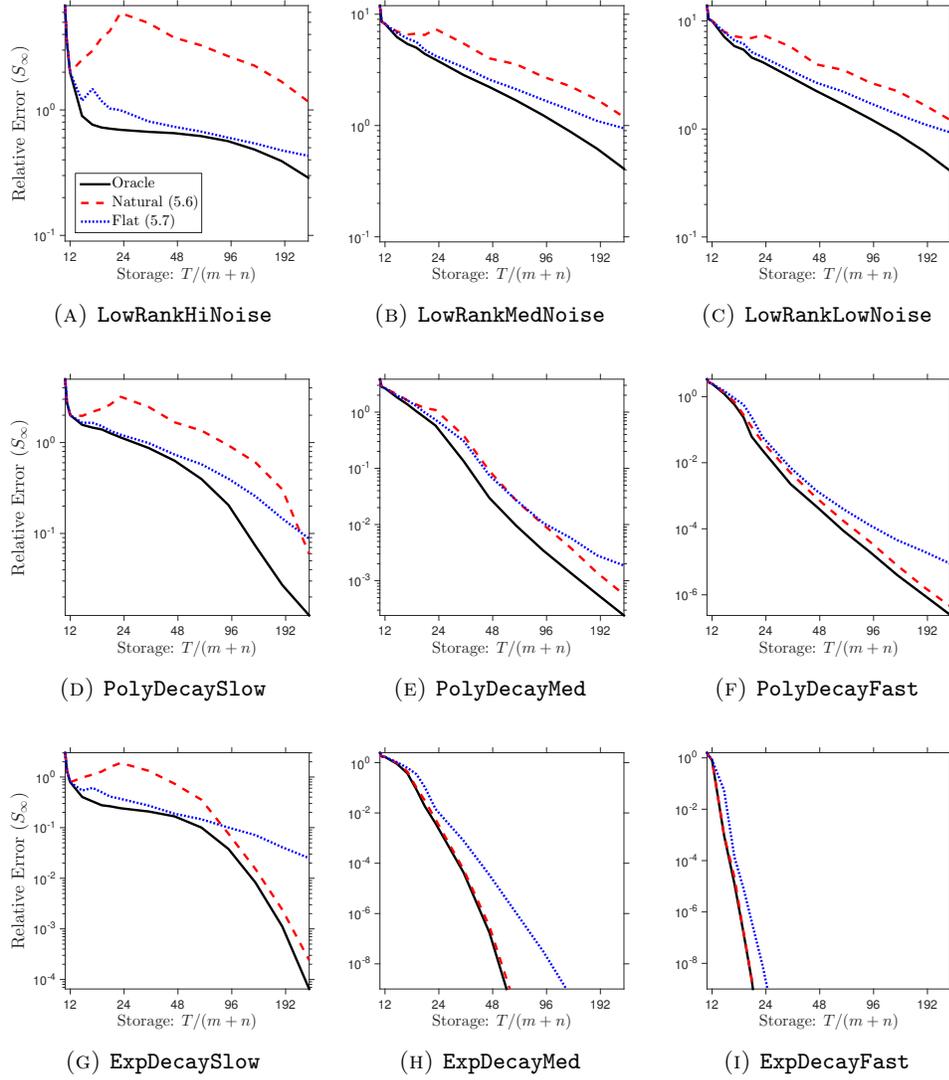


FIG. SM7: **Relative error for proposed method with *a priori* parameters.** (Gaussian maps, effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

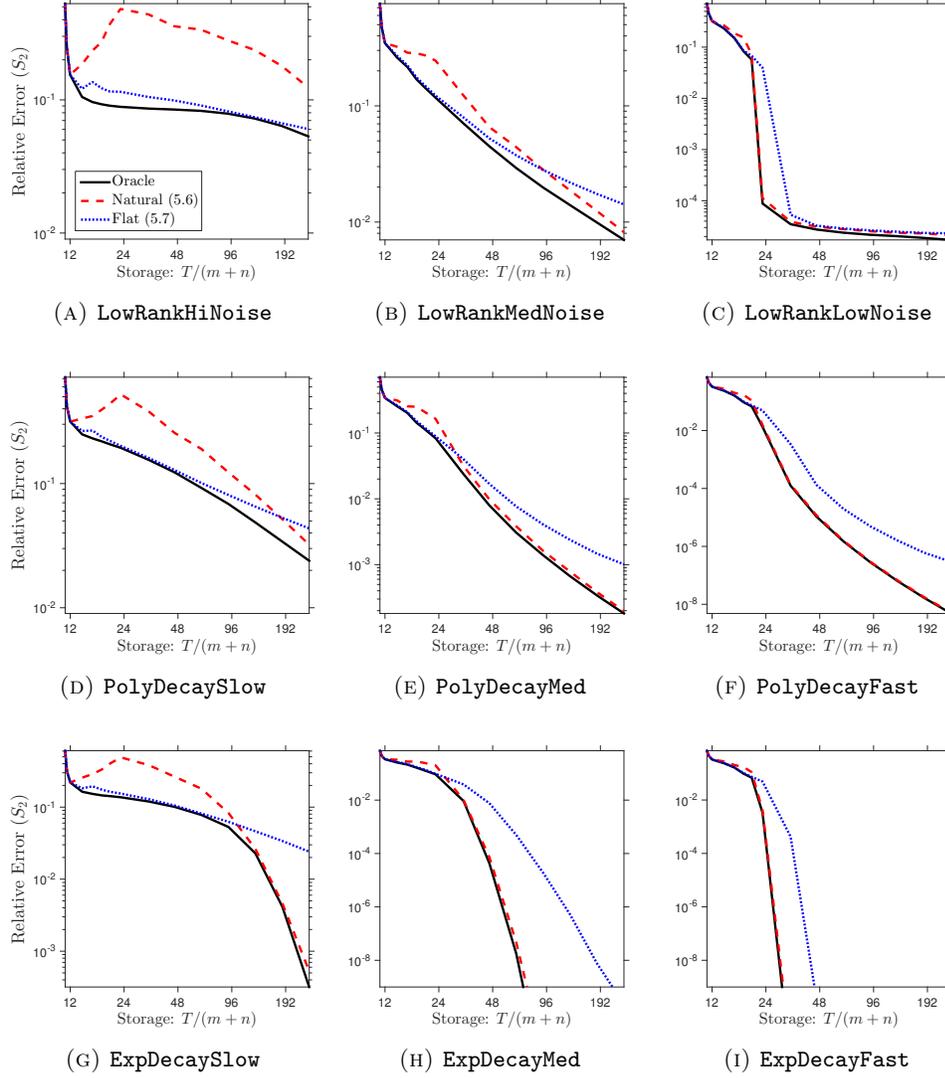


FIG. SM8: **Relative error for proposed method with *a priori* parameters.** (Gaussian maps, effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

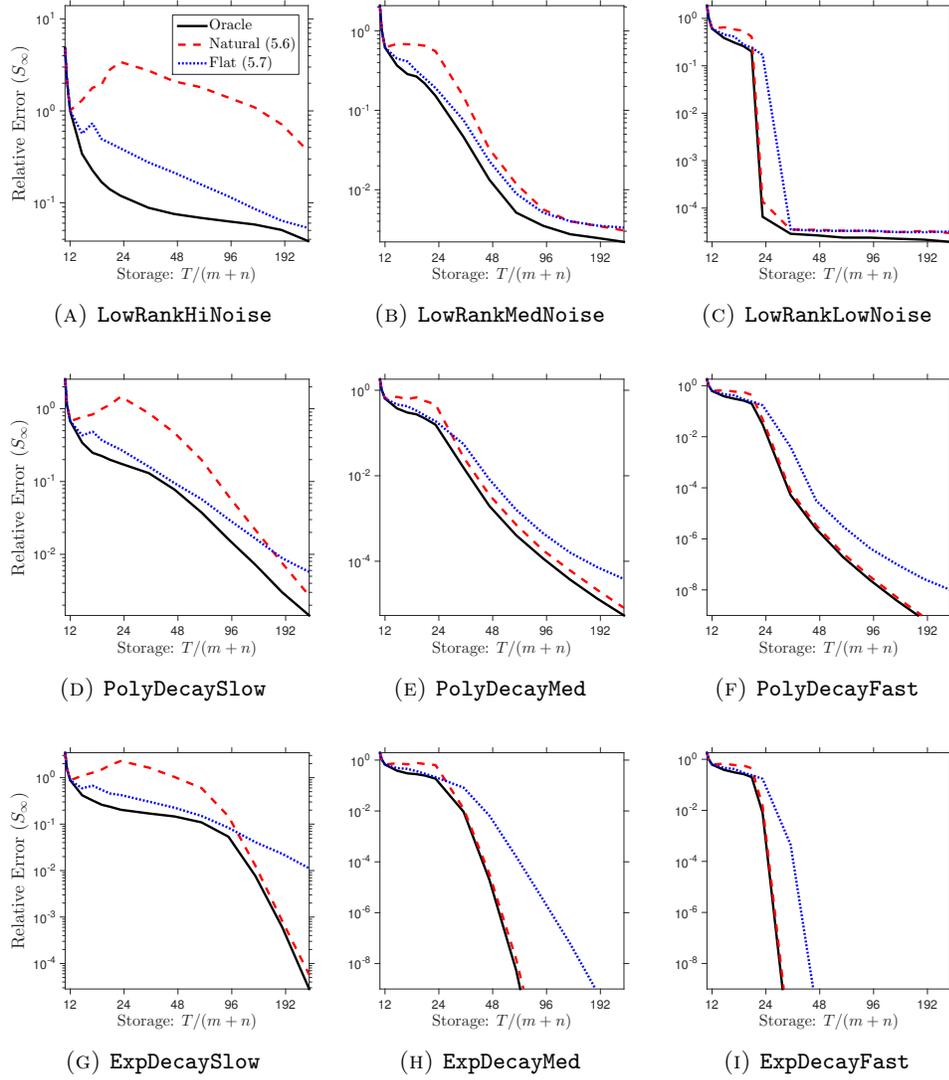


FIG. SM9: **Relative error for proposed method with *a priori* parameters.** (Gaussian maps, effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle performance of the proposed fixed-rank approximation (2.10) with its performance at theoretically justified parameter values. See subsection 6.5 for details.

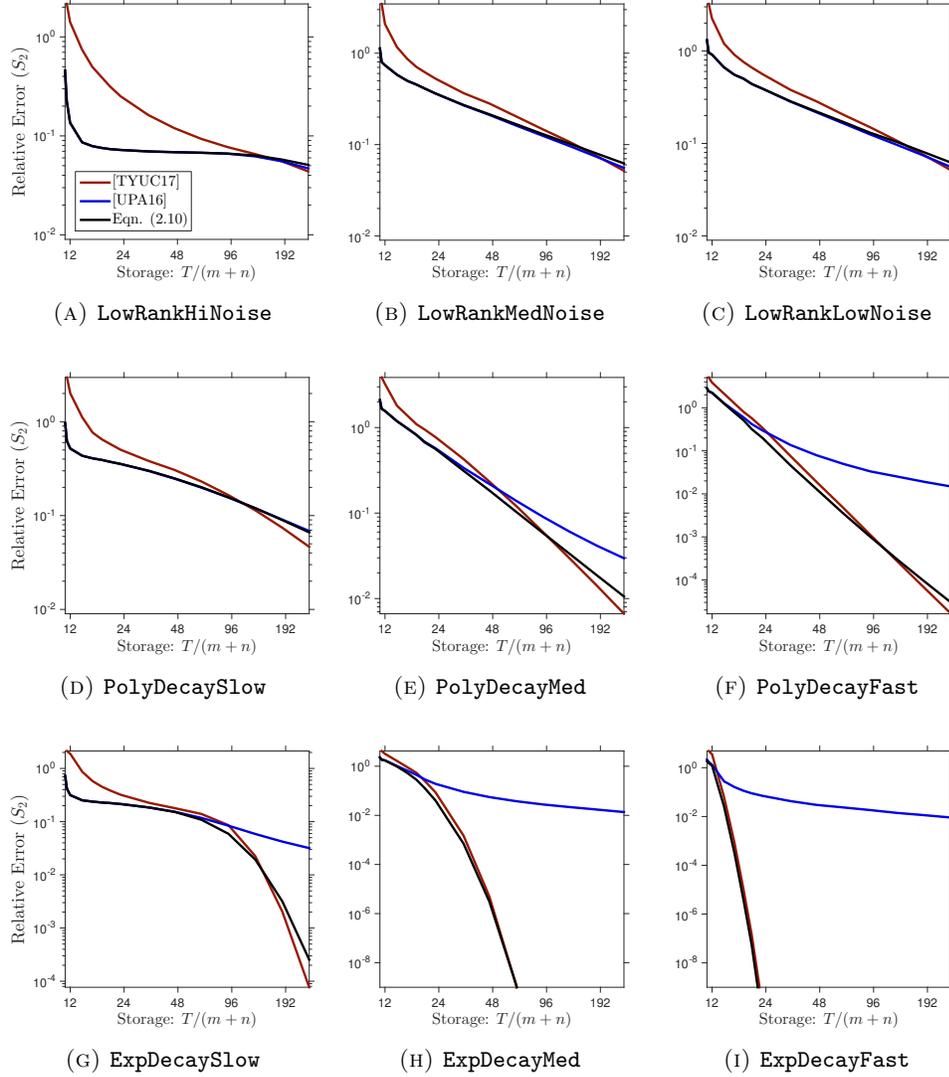


FIG. SM10: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See [subsection 6.2.2](#) for details.

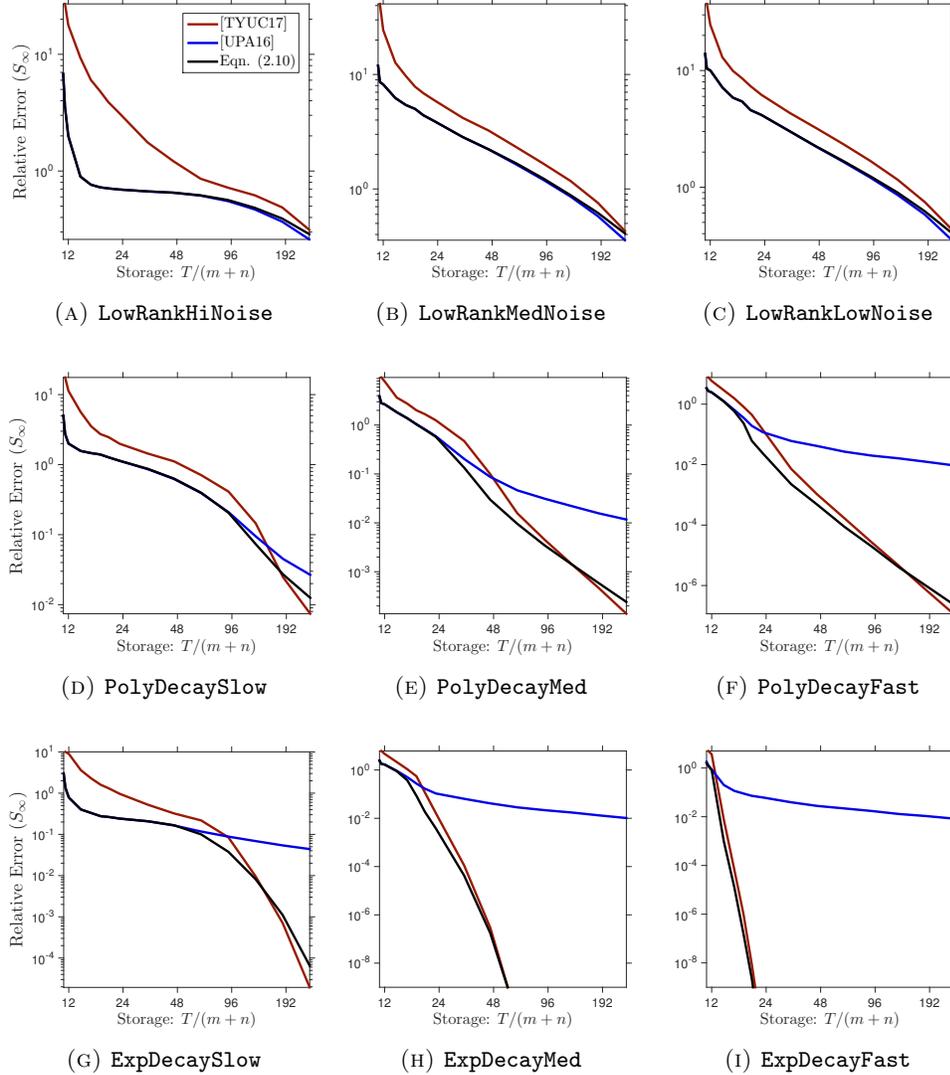


FIG. SM11: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 5$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See subsection 6.2.2 for details.

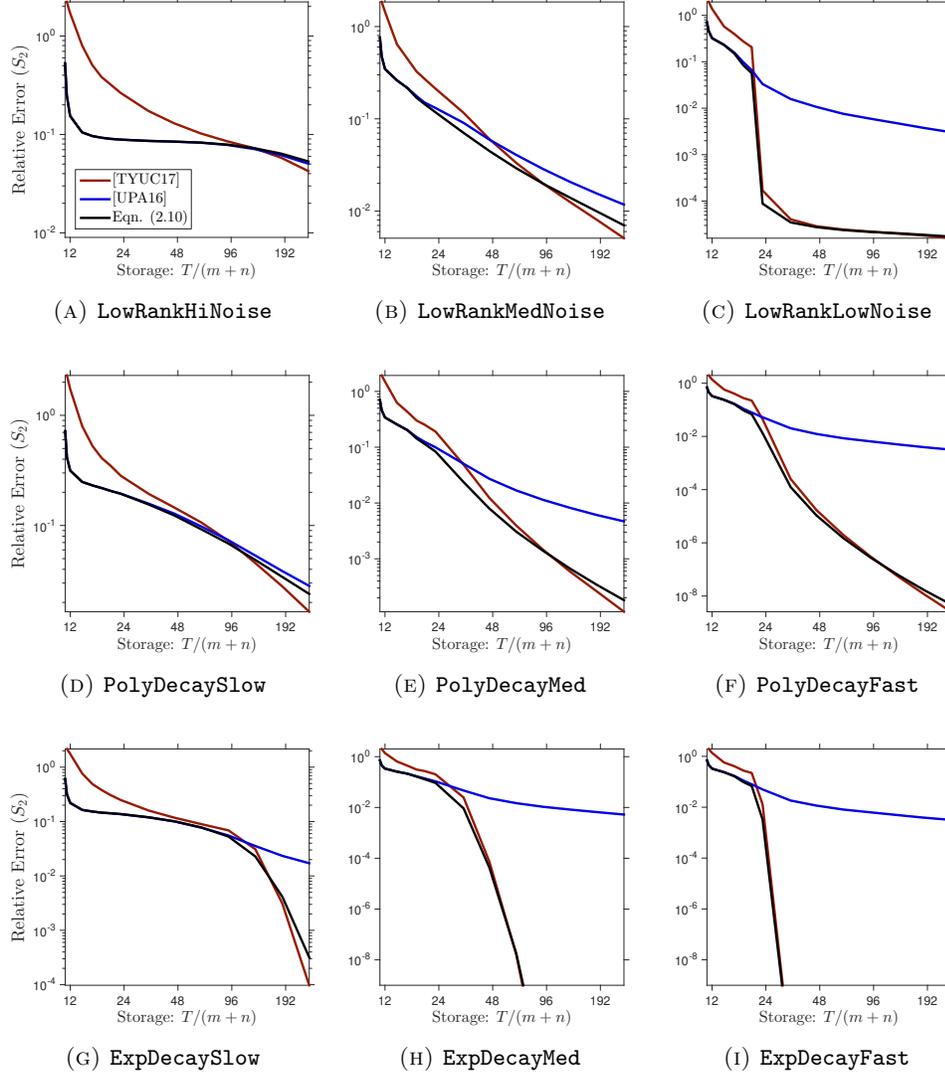


FIG. SM12: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten 2-norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See subsection 6.2.2 for details.

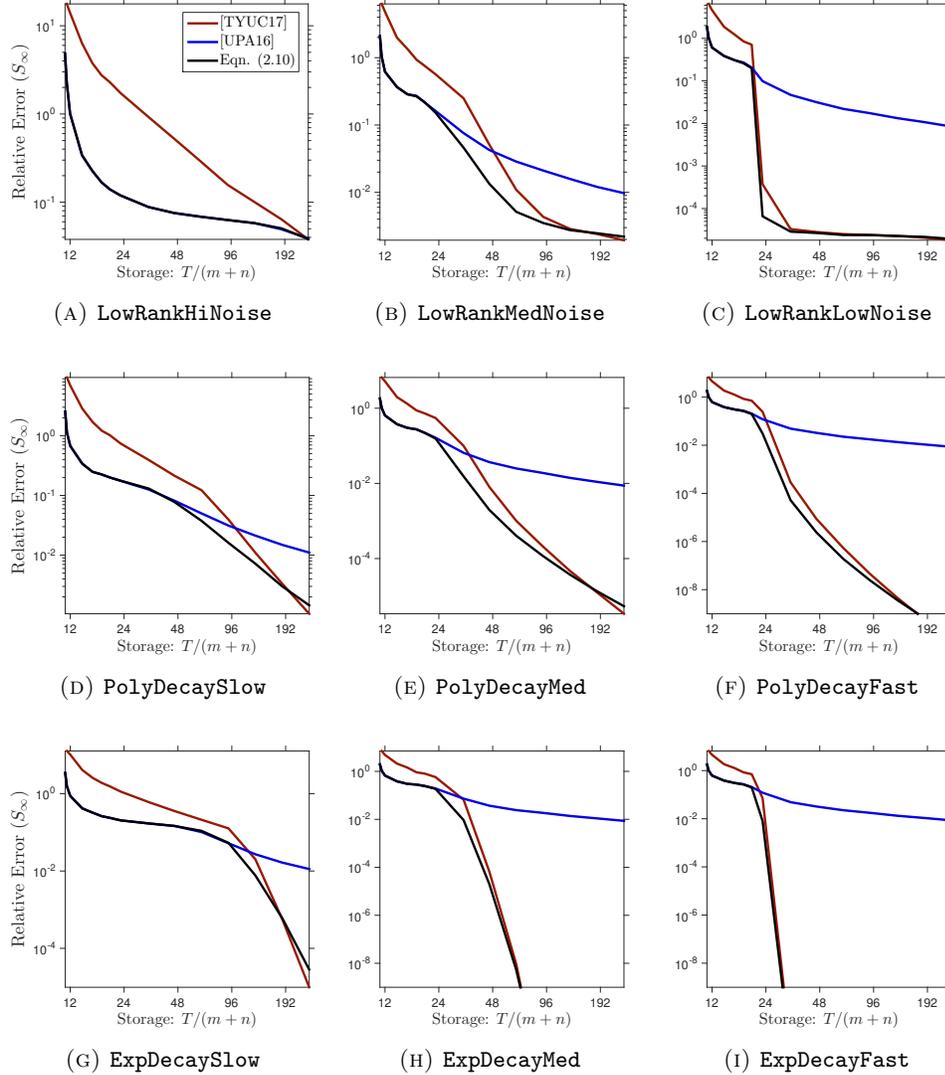


FIG. SM13: **Comparison of reconstruction formulas: Synthetic examples.** (Gaussian maps, effective rank  $R = 20$ , approximation rank  $r = 10$ , Schatten  $\infty$ -norm.) We compare the oracle error achieved by the proposed fixed-rank approximation (2.10) against methods (6.1) and (6.2) from the literature. See subsection 6.2.2 for details.

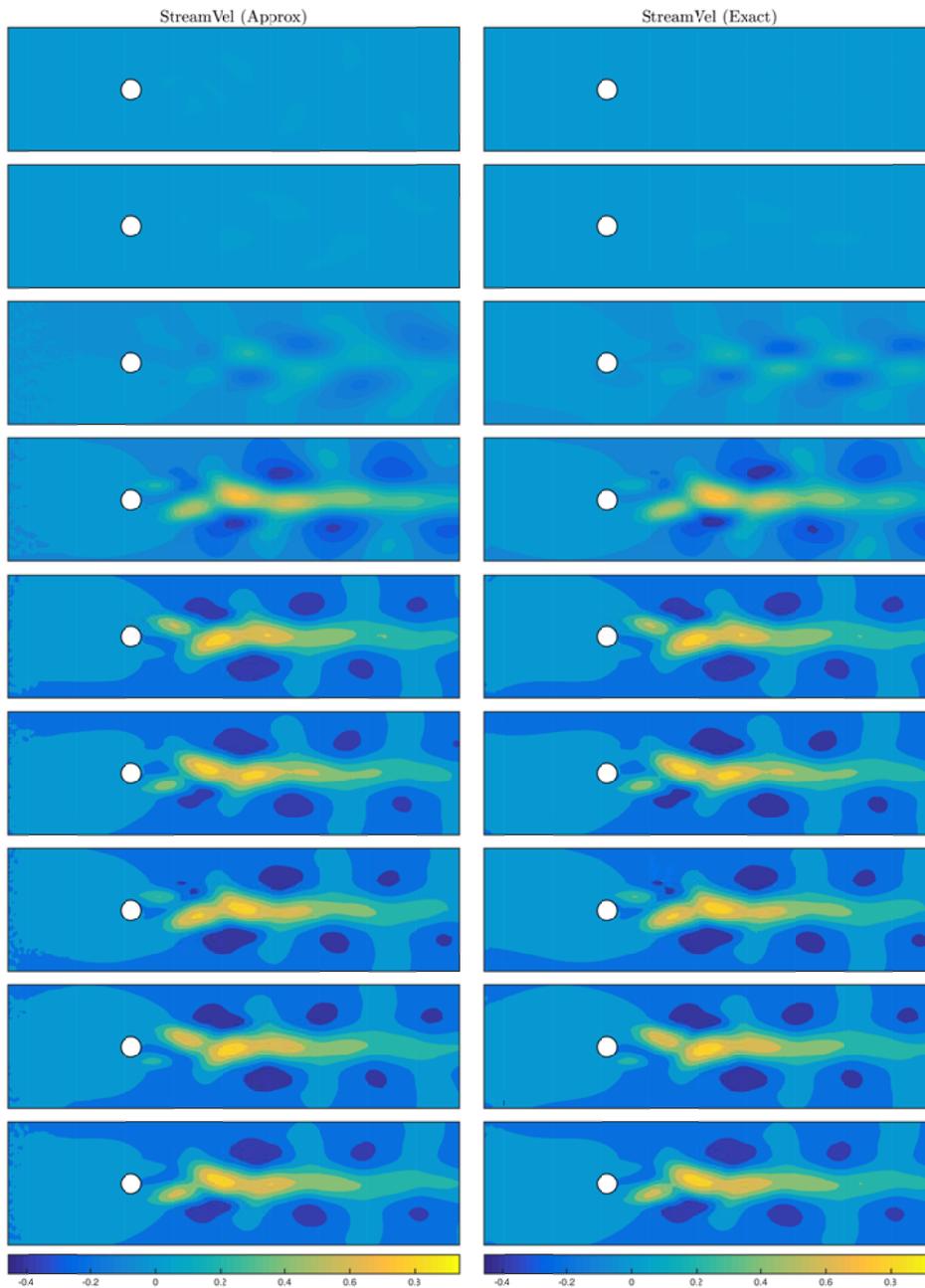


FIG. SM14: **Approximation of StreamVel.** (Sparse maps, approximation rank  $r = 10$ , storage budget  $T = 48(m + n)$ .) The columns of the matrix **StreamVel** describe the fluctuations of the streamwise velocity field about its mean value as a function of time. From top to bottom, the panels show columns 1, 501, 1001, 1501, 2001, 2501, 3001, 3501, 4001. The **left-hand side** displays the approximation of the flow field, and the **right-hand side** displays the exact flow field. The heatmap indicates the magnitude of the fluctuation. See [subsection 6.8](#) for details.