

# Greedy Sensor Placement with Cost Constraints

Emily Clark, Travis Askham, Steven L. Brunton, *Member, IEEE*, J. Nathan Kutz, *Member, IEEE*

**Abstract**—The problem of optimally placing sensors under a cost constraint arises naturally in the design of industrial and commercial products, as well as in scientific experiments. We consider a relaxation of the full optimization formulation of this problem and then extend a well-established greedy algorithm for the optimal sensor placement problem without cost constraints. We demonstrate the effectiveness of this algorithm on data sets related to facial recognition, climate science, and fluid mechanics. This algorithm is scalable and often identifies sparse sensors with near-optimal reconstruction performance, while dramatically reducing the overall cost of the sensors. We find that the cost-error landscape varies by application, with intuitive connections to the underlying physics. Additionally, we include experiments for various pre-processing techniques and find that a popular technique based on the singular value decomposition is often suboptimal.

## I. INTRODUCTION

**D**ETERMINING the optimal placement of sensors under a cost constraint is relevant to many fields of scientific research and industry. Indeed, such considerations are critical in evaluating global monitoring systems and characterizing spatio-temporal dynamics (e.g. the brain, ocean and atmospheric dynamics, power grid networks, fluid flows, etc). For these applications, it is typical that only a limited number of measurements can be made of the system due to either prohibitive expense (i.e. either sensors are expensive, or they are expensive to place, or both) or the inability to place a sensor in a desired location (inaccessibility).

There are various high-level objectives for sensor placement, most of which are well studied. Common objectives include classification [1], [2], reconstruction [3]–[9], reduced-order modeling [8], [10]–[12], and control [13], [14]. In contrast, the case in which different sensor locations may have different associated costs has received relatively little attention, with the notable exception of the submodularity literature [14]–[22]. As reconstruction error is not itself submodular (see Section I-B below for more on submodularity), the goal of this text is to develop algorithms that directly target the reconstruction objective, while incorporating heterogeneous constraints on the measurement locations.

To this end, we develop a heuristic, greedy sampling strategy whereby the sensor placement optimization is formulated as a cost-constrained problem in a relaxed form. We further introduce a parameter representing the balance between the quality of the reconstruction and the cost, and thus can evaluate

a cost-error curve. The simple algorithmic structure proposed is based on a modification of the pivoted QR decomposition algorithm of Businger and Golub [23], which represents a matrix as the product of an orthogonal matrix,  $\mathbf{Q}$ , and an upper-triangular matrix,  $\mathbf{R}$ , and provides an effective and scalable strategy for economical sensor placement for a wide range of scientific and engineering applications<sup>1</sup> [9], [12].

### A. Sensor placement for reconstruction

Since our focus is on full-state reconstruction, here we consider the problem of finding sensor locations that minimize the reconstruction error. In principle, the map from the measurements of a system to the reconstruction can take any form. Here, we will review those methods for which the reconstruction is obtained by applying a *linear* map to the values at the sensors; this can be viewed as an interpolation problem, where the sensors are interpolation points. There are a few reasons to restrict the search to linear maps: (i) it is easy to check the stability and optimality of a given set of points, (ii) it is straightforward to design efficient algorithms for sensor placement, and, (iii) as noted above, the sensor locations can be interpreted as interpolation points.

A brute-force solution of the sensor placement problem may be obtained by searching over all possible subsets of the sensors, but this approach quickly becomes intractable, as the number of subsets increases combinatorially. However, randomly placed sensors perform surprisingly well. For instance, Wright et al. observed that, given a generic basis in which samples of the signal will be sparse, it is possible to perfectly reconstruct a signal that has been downsampled or randomly projected [24]. The compressed sensing literature provides a theoretical basis for the surprising effectiveness of random, or rather *incoherent*, measurements in this setting; see, inter alia, [25]–[29]. Such an approach does not necessarily make use of any full-state observations of the system (though some model for the system is implied), and random sensors have been observed to be less efficient than sensors that take this data into account [9].

A common data-driven approach is to start with a *tailored basis* derived from the observed samples, typically given by the dominant singular vectors [9]–[11], [30]. See [7], [8] for early examples of signal reconstruction from a limited number of sensors using such a basis. A number of heuristic choices for the locations have been developed, including placing sensors at the extrema of the singular vectors [4]–[6].

Before continuing our discussion of sensor placement in a tailored basis, we require some notation. Assume we have  $m$

E. Clark is with the Department of Physics at the University of Washington, Seattle, WA, 98195-1560 USA email: eclark7@uw.edu

S. Brunton is with the Department of Mechanical Engineering at the University of Washington, Seattle, WA, 98195-2600 USA email: sbrunton@uw.edu

T. Askham and J. N. Kutz are with the Department of Applied Mathematics at the University of Washington, Seattle, WA, 98195-3925 USA email: askham@uw.edu and kutz@uw.edu

<sup>1</sup>MATLAB code for our algorithm can be found at <https://github.com/askhamwhat/sensors-cost-paper>, along with a simple example of its use and codes to make some of the figures from Section VI.

samples of data  $\mathbf{x}_i \in \mathbb{R}^n$ , and let these samples form the rows of a matrix  $\mathbf{X}$ :

$$\mathbf{X} = \begin{pmatrix} \text{--- } \mathbf{x}_1 \text{ ---} \\ \text{--- } \mathbf{x}_2 \text{ ---} \\ \vdots \\ \text{--- } \mathbf{x}_m \text{ ---} \end{pmatrix}. \quad (1)$$

Thus the columns of  $\mathbf{X}$  correspond to spatial locations. Furthermore, let  $\Psi$  be some matrix derived from  $\mathbf{X}$  (e.g.  $\Psi$  may be taken to be the right singular vectors of  $\mathbf{X}$ , random linear combinations of the rows of  $\mathbf{X}$ , or  $\mathbf{X}$  itself). For an index set  $J$  and any matrix  $\mathbf{M}$ , let  $\mathbf{M}_{\cdot J}$  denote the matrix formed by the columns of  $\mathbf{M}$  with index in  $J$ . If the size of  $J$  is fixed, it is known that the set of indices  $\hat{J}$  that maximizes the product of the singular values of  $\Psi_{\cdot J}$  (or equivalently maximizes the determinant of  $\Psi_{\cdot J}$ ) provides optimal interpolation points for  $\Psi$  [31], [32] (see Theorem 1 below for the definition of optimal).

The problem of finding such a  $\hat{J}$  is nonconvex and NP-hard, but there are reasonable approximate algorithms. Gu and Eisenstat developed a polynomial time algorithm for computing  $J$  when the optimality criterion is relaxed slightly [31]. In [33], Li et al. considered the related problem of dual volume sampling, which would allow the placement of more sensors than there are samples of data, using a probabilistic framework. Joshi and Boyd reformulated determinant maximization as an approximate convex problem, which may be solved in polynomial time and is observed to provide nearly optimal sensors [3]. While these approaches scale polynomially in the number of sensors and the size of the data, they are not as computationally efficient as some of the existing greedy algorithms for interpolation, especially for high-dimensional data. Further, the examples on which the greedy algorithms are known to fail appear to be pathological, i.e. it is incredibly unlikely that the greedy approach will fail in practice.

The greedy sensor selection algorithm which is of greatest interest in this paper is based on the column pivoted QR decomposition. In particular, for a given number of sensors  $k$ , one simply selects  $J$  to be the first  $k$  column pivots of  $\Psi$  (see Section II for an explanation as to why this is a greedy approach for maximizing the product of the singular values of  $\Psi_{\cdot J}$ ). This algorithm is the basis for practical approaches to computing the interpolative decomposition [32], [34], which is commonly used to compress low-rank matrices. The algorithm is also used in the discrete empirical interpolation method (DEIM) from reduced order modeling [10], [11], in its more stable Q-DEIM formulation [12]. For high-dimensional problems with many samples of data, standard techniques from the burgeoning field of randomized algorithms for linear algebra may be used to improve the efficiency of these schemes [35], [36].

### B. Submodularity

In a line of research parallel to the linear-algebra-based methods described above, the sensor placement problem is often formulated as submodular function optimization. A set function is submodular if it obeys a diminishing returns property: adding an element to a set produces a greater change

in the function than adding the same element to a superset of that set. Greedy methods for optimizing submodular functions are well studied; see [37] for a proof of the near-optimal performance of greedy methods for submodular functions and, inter alia, [15]–[20], [38] for further developments. In particular, [15] provides a comprehensive review of submodular functions and the maximization thereof.

The topic is of interest in this work because certain performance metrics for sensor placement — including the determinant maximizing metric used in this paper — are submodular, which means that greedy methods enjoy some optimality guarantees. For example, [39], [40] leverage the submodularity of the mutual information between chosen and unchosen sensor locations to find near-optimal observation locations to model Gaussian processes. The FrameSense algorithm developed in [18] is a fast greedy method that minimizes the frame potential of the basis matrix  $\Psi$ , a measure of the orthogonality of its rows. Under certain assumptions on the structure of the linear model, maximizing the frame potential implies a near-optimal bound on the mean square error of parameter recovery. Submodularity-based methods can be extended to include a cost function on sensor location, as in [16], [21], [22].

The related problem of controlling a system based on a limited number of measurements and actuators has also been thoroughly explored, as in [41]–[44]. Furthermore, it can be shown that certain controllability and observability metrics are submodular, and thus near-optimal sensor and actuator locations can be chosen using simple greedy algorithms [13], [45]–[47]. The submodularity properties are extended to include a heterogeneous cost function in [14].

### C. Contributions of this work

It can be verified through numerical experiment that the reconstruction error of a matrix after interpolation is not itself submodular in the interpolation points. However, the log-determinant is a good proxy for reconstruction, see Theorem 1, and is submodular. While the standard optimality guarantees either do not apply or are significantly weakened for a possibly decreasing submodular function like the log-determinant, the method is empirically effective in the unconstrained reconstruction setting, and the QR decomposition approach for computing the sensor locations is efficient and scalable. Therefore, we have set out in this work to adapt the QR decomposition approach to the cost-constrained setting.

In the techniques described in Section I-A, an optimal map and set of sensor locations are found for a fixed number of sensors. This is equivalent to the cost-constrained sensor placement problem when each sensor has the same cost. In the case that some sensor locations should be entirely excluded, corresponding to an infinite cost, i.e. an inaccessible measurement location, and the remaining locations are of uniform cost, again the algorithms above may be used by simply narrowing the search to the allowed sensor locations (note that such a restriction has implications for the stability of the interpolation map).

We show that it is simple to modify the pivoted-QR-decomposition-based scheme to incorporate a cost constraint

for problems in between these extremes, i.e. for problems in which some sensor locations cost more than others but may be more informative. The method is obtained by writing the cost-constrained problem in a relaxed form, which introduces a parameter representing the balance between the quality of the reconstruction and the cost, and then varying that parameter to trace out a cost-error curve. For each value of the parameter, we use a greedy algorithm to add sensor locations one by one.

We test the performance of our method on data sets from facial recognition, climate science, and fluid mechanics using a standard training-set/testing-set apparatus. In many cases, the proposed algorithm displays a significant advantage over methods based on randomly selected sensors. We also compare with known performance bound approximations where applicable and find that our algorithm is often near the approximately optimal solution.

Furthermore, we include a brief discussion of the effect of data pre-processing (i.e. the choice of  $\Psi$ ) on the quality of the sensor locations; in particular, we compare the performance when applied to the raw data, the first several singular vectors of the data, and randomized projections of the data. We find that with the unmodified column pivoted QR decomposition, using the raw data (taking  $\Psi = \mathbf{X}$ ) yields the lowest reconstruction error at a given number of sensors, while randomized projections of the data lead to slightly higher errors at a reduced computational cost. Taking the first several singular vectors of  $\mathbf{X}$  gives the highest reconstruction errors of the three pre-processing methods tested. The results are the same when we incorporate a cost function using our modified QR decomposition, but additionally we note that all three pre-processing techniques yield comparable costs.

The remainder of this paper is organized as follows. In Section II, we summarize some relevant results from the interpolation literature and present our problem formulation. We then develop an algorithm for sensor placement in Section III which is a simple extension of the existing methods. In Section IV, we describe the three data sets we use as examples. We discuss pre-processing in Section V. We then apply our method to the three data sets and discuss the performance in Section VI. Finally, we provide some concluding remarks and indicate possible future avenues for research in Section VII.

## II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we fix some notation, formulate the linear sensor placement problem with non-uniform cost constraints, and review the pivoted QR algorithm, which will form the basis of our method.

### A. Setting, notation, and problem formulation

Let  $\mathbf{x}^i \in \mathbb{R}^n$  denote samples of some system and let  $\boldsymbol{\eta} \in \mathbb{R}_+^n$  denote non-negative costs associated with each sample location. We collect the samples  $\mathbf{x}^i$  as the rows of a matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$ . The sensor placement problem with cost constraints then seeks out an optimal subset  $\hat{J}$  of the column indices of  $\mathbf{X}$  which balances the associated cost  $\sum_{j \in \hat{J}} \eta_j$  with reconstruction error and stability, which we define below.

For a given set of indices,  $J = \{j_1, \dots, j_l\}$ , it is simple to construct the optimal linear map for reconstructing the entries in  $\mathbf{X}$ . Let

$$\hat{\mathbf{T}}(J) = \arg \min_{\mathbf{T} \in \mathbb{R}^{l \times n}} \|\mathbf{X} - \mathbf{X}_{\cdot J} \mathbf{T}\|_F, \quad (2)$$

where  $\mathbf{X}_{\cdot J}$  denotes the matrix given by collecting the columns of  $\mathbf{X}$  whose indices are in  $J$ , and  $\|\cdot\|_F$  denotes the Frobenius norm. It is well known that  $\hat{\mathbf{T}}(J) = \mathbf{X}_{\cdot J}^\dagger \mathbf{X}$ , where  $\mathbf{X}_{\cdot J}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{X}_{\cdot J}$ , i.e. this is the least-squares solution. Therefore, the relative reconstruction error for linear sensor placement is given by

$$e(J) = \frac{\|\mathbf{X} - \mathbf{X}_{\cdot J} \mathbf{X}_{\cdot J}^\dagger \mathbf{X}\|_F}{\|\mathbf{X}\|_F}, \quad (3)$$

and the stability of the interpolation map is determined by  $\|\mathbf{X}_{\cdot J}^\dagger \mathbf{X}\|_\infty$ . In the following, we will focus on computing a subset  $J$  such that the error is small and the map is stable.

We will use much of the notation introduced above throughout the paper. When necessary, we will denote data matrices and errors corresponding to a training set by  $\mathbf{X}^{\text{tr}}$  and  $e^{\text{tr}}(J)$  and the matrices and errors corresponding to a testing set by  $\mathbf{X}^{\text{te}}$  and  $e^{\text{te}}(J)$ . Note that

$$e^{\text{te}}(J) = \frac{\|\mathbf{X}^{\text{te}} - \mathbf{X}_{\cdot J}^{\text{te}} \mathbf{X}_{\cdot J}^{\text{tr} \dagger} \mathbf{X}^{\text{tr}}\|_F}{\|\mathbf{X}^{\text{te}}\|_F}, \quad (4)$$

i.e. the operator  $\hat{\mathbf{T}}(J)$  is always determined by the training set. It is also common (see [9], [11], [35], [36]) to reduce the computational cost associated with finding  $J$  by applying the algorithm to an  $r \times n$  matrix  $\Psi^{\text{tr}}$  which captures the dominant features of  $\mathbf{X}^{\text{tr}}$  for some  $r \ll m$ , e.g. to a matrix of singular vectors of  $\mathbf{X}^{\text{tr}}$  or a matrix given by random linear combinations of the rows of  $\mathbf{X}^{\text{tr}}$ . In this case, we define  $\hat{\mathbf{T}}(J) = \Psi_{\cdot J}^{\text{tr} \dagger} \Psi^{\text{tr}}$ .

It is now possible to define the linear sensor placement problem with cost constraints. Let  $\boldsymbol{\eta}$  be the cost vector as described above and let  $s$  and  $b$  denote desired upper bounds on the stability of the map  $\hat{\mathbf{T}}(J)$  and the budget, respectively. Then we may write the cost-constrained problem as

$$\hat{J} = \arg \min_J e(J) \text{ s.t. } \sum_{j \in J} \eta_j \leq b \text{ and } \|\hat{\mathbf{T}}(J)\|_{\infty, \text{vec}} \leq s, \quad (5)$$

where  $\|\cdot\|_{\infty, \text{vec}}$  denotes the maximum absolute value over the entries of a matrix. Our algorithm will actually focus on the following relaxation of (5). Note that, for a given  $b$ , there exists a  $\lambda$  such that the problem

$$\hat{J} = \arg \min_J e(J) + \lambda \sum_{j \in J} \eta_j \text{ s.t. } \|\hat{\mathbf{T}}(J)\|_{\infty, \text{vec}} \leq s \quad (6)$$

and (5) have the same solution. Because we are often interested in the cost-error landscape, we seek the solution of (6) for a number of values of  $\lambda$ , so that we trace out a cost-error curve. We note that the main algorithm we present in Section III does not actually solve (6). Instead, we seek a greedy approximate solution which does not strictly enforce the stability constraint

but uses a heuristic strategy to bias the sensors in favor of stability.

In the case that all entries of  $\boldsymbol{\eta}$  are equal and positive, the constraint  $\sum_{j \in J} \eta_j \leq b$  simplifies to the constraint  $|J| \leq b/\eta_1$ . This is equivalent to the problem of optimally placing a specified number of sensors, a well-studied problem which we briefly review in the remainder of this section.

### B. Theoretical results

In this section, we will motivate determinant maximization algorithms (sometimes called D-optimal) for selecting a fixed number of interpolation points, i.e. methods that maximize the product of the singular values of  $\mathbf{X}_{\cdot J}^T$ . As observed in [32], the strong rank revealing QR decomposition methods of [31] provide a polynomial time algorithm for computing an approximately determinant-maximizing subset  $J$  such that the error  $e(J)$  is near-optimally small and the map  $\hat{\mathbf{T}}(J)$  is near-optimally stable. The optimal performance is summarized in the following theorem.

*Theorem 1:* (from [32]) Let  $\mathbf{X} \in \mathbb{R}^{m \times n}$  and let  $k \leq l = \min(m, n)$ . Then, there exists a  $J$  such that  $|J| = k$ ,

$$\|\mathbf{X} - \mathbf{X}_{\cdot J} \hat{\mathbf{T}}(J)\|_F \leq \sqrt{1 + k(l - k)} \sum_{j=k+1}^l \sigma_j(\mathbf{X}), \quad (7)$$

where  $\sigma_j(\mathbf{X})$  denotes the  $j$ th singular value of  $\mathbf{X}$ , and

$$\|\hat{\mathbf{T}}(J)\|_F \leq \sqrt{k(n - k) + k}. \quad (8)$$

Remarkably, relaxing the bounds to

$$\|\mathbf{X} - \mathbf{X}_{\cdot J} \hat{\mathbf{T}}(J)\|_F \leq \sqrt{1 + lk(l - k)} \sum_{j=k+1}^l \sigma_j(\mathbf{X}), \quad (9)$$

and

$$\|\hat{\mathbf{T}}(J)\|_F \leq \sqrt{nk(n - k) + k}, \quad (10)$$

the algorithms of [31] compute such a  $J$  using, typically,  $O(mnk)$  flops and at most  $O(mnl)$  flops.

While provably near-optimal and polynomial time algorithms are provided by [31], the standard QR algorithm with column pivoting is much more efficient and scalable and tends to achieve similar bounds. Because the column pivoting decision is equivalent to the greedy heuristic for the log-determinant, which is a submodular function, one may try to appeal to the standard optimality bounds [37] and bounds for the cost-constrained case [16]. Unfortunately, the log-determinant can decrease for large classes of matrices, including those considered in this paper, and the existing bounds are either weakened or no longer apply for functions which can decrease. In particular, a statement about the reconstruction error along the lines of Theorem 1 is not possible and there are indeed known failure cases (see, for instance, the Kahan matrix example of [31]). Fortunately, such failures appear to be limited to pathological examples.

### C. QR with column pivoting for sensor placement

In this section, we review the column-pivoted QR decomposition algorithm of [23], which can be viewed as a greedy determinant-maximizing method for sensor placement without cost constraints. This algorithm is the basis for our cost-constrained method.

The method of [23] is based on the repeated application of Householder reflectors to triangularize a given matrix. Let a vector  $\mathbf{v} \in \mathbb{R}^m$  be given. We can then define a Householder reflector [48] which maps  $\mathbf{v}$  to  $\|\mathbf{v}\|_2 \mathbf{e}^1$ , where  $\mathbf{e}^1$  is the first standard basis vector in  $\mathbb{R}^m$ . Let  $\sigma = \|\mathbf{v}\|_2$  and let  $v_1$  denote the first entry of  $\mathbf{v}$ . Then, the matrix

$$\mathbf{H}(\mathbf{v}) := \mathbf{I} - \frac{(\mathbf{v} + \text{sign}(v_1)\sigma\mathbf{e}^1)(\mathbf{v} + \text{sign}(v_1)\sigma\mathbf{e}^1)^T}{\sigma(\sigma + |v_1|)} \quad (11)$$

maps  $\mathbf{v}$  to  $-\text{sign}(v_1)\sigma\mathbf{e}^1$ . It is simple to verify that there exists a  $\mathbf{u} \in \mathbb{R}^m$  of unit norm such that the map is given by  $\mathbf{I} - 2\mathbf{u}\mathbf{u}^T$ , which is the standard form of a Householder reflector. This expression also makes it clear that  $\mathbf{H}(\mathbf{v})$  is its own inverse.

Using the notation of [31], the standard QR decomposition algorithm with column pivoting applied to a matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$  proceeds by iteratively defining an orthogonal matrix  $\mathbf{Q}^k$  and a permutation of the indices  $J^k = \{j_1^k, \dots, j_n^k\}$  such that

$$\mathbf{X}\mathbf{P}(J^k) = \mathbf{Q}^k \mathbf{R}^k, \quad \mathbf{R}^k = \begin{pmatrix} \mathbf{A}^k & \mathbf{B}^k \\ 0 & \mathbf{C}^k \end{pmatrix}, \quad (12)$$

where  $\mathbf{A}^k \in \mathbb{R}^{k \times k}$  is upper triangular,  $\mathbf{B}^k \in \mathbb{R}^{k \times (n-k)}$ ,  $\mathbf{C}^k \in \mathbb{R}^{(m-k) \times (n-k)}$ , and

$$\mathbf{P}(J^k) = \begin{pmatrix} \mathbf{e}^{j_1^k} & \mathbf{e}^{j_2^k} & \dots & \mathbf{e}^{j_n^k} \end{pmatrix}. \quad (13)$$

For  $k < \min(m, n)$ , the decomposition (12) is sometimes called a *partial QR factorization*.

Let  $J^0 = \{1, \dots, n\}$  and  $\mathbf{Q}^0 = \mathbf{I}$ . Denote the columns of  $\mathbf{C}^k$  by  $\mathbf{c}^{k,i}$  for  $i = 1, \dots, n - k$ . To obtain the  $k + 1$ st iterate from the  $k$ th, let  $\mathbf{c}^{k,l}$  be the column of  $\mathbf{C}^k$  with the largest norm. We update the permutation indices to move this column to the front, i.e.  $j_{k+1}^{k+1} = j_{k+l}^k$ ,  $j_{k+l}^{k+1} = j_{k+1}^k$ , and  $j_i^{k+1} = j_i^k$  for all other  $i$ . If we update the orthogonal matrix  $\mathbf{Q}^k$  via

$$\mathbf{Q}^{k+1} = \mathbf{Q}^k \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{H}(\mathbf{c}^{k,l}) \end{pmatrix}, \quad (14)$$

then

$$\mathbf{X}\mathbf{P}(J^{k+1}) = \mathbf{Q}^k \begin{pmatrix} \mathbf{A}^{k+1} & \mathbf{B}^{k+1} \\ 0 & \mathbf{C}^{k+1} \end{pmatrix}, \quad (15)$$

where  $\mathbf{A}^{k+1}$ ,  $\mathbf{B}^{k+1}$  and  $\mathbf{C}^{k+1}$  are of the correct form and  $\mathbf{A}_{k+1, k+1}^{k+1} = \pm \|\mathbf{c}^{k,l}\|$ .

The idea of using the column-pivoted QR algorithm for sensor placement is that, for a given  $k$ , the first  $k$  pivots  $j_1^k, \dots, j_k^k$  should be a good choice of sensor locations. Let  $J = \{j_1^k, \dots, j_k^k\}$ . Many of the quantities of interest in reconstruction can then be determined using the schematic (12). Note that

$$\mathbf{X}_{\cdot J} = \mathbf{Q}^k \begin{pmatrix} \mathbf{A}^k \\ 0 \end{pmatrix} \quad \text{so that} \quad \mathbf{X}_{\cdot J}^\dagger = \left( (\mathbf{A}^k)^{-1} \quad 0 \right) (\mathbf{Q}^k)^\top.$$

Then the interpolation matrix is given by

$$\hat{\mathbf{T}}(J)\mathbf{P}(J^k) = \mathbf{X}_{\cdot J}^\dagger \mathbf{X} \mathbf{P}(J^k) = \begin{pmatrix} \mathbf{I} & (\mathbf{A}^k)^{-1} \mathbf{B}^k \end{pmatrix}. \quad (16)$$

Finally, the error can be found by observing that

$$\mathbf{X} \mathbf{P}(J^k) - \mathbf{X}_{\cdot J} \hat{\mathbf{T}}(J) \mathbf{P}(J^k) = \mathbf{Q}^k \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{C}^k \end{pmatrix},$$

so that  $e(J) = \|\mathbf{C}^k\|_F / \|\mathbf{X}\|_F$ . Therefore, both the stability of the map and the error are determined by the factors  $\mathbf{A}^k$ ,  $\mathbf{B}^k$ , and  $\mathbf{C}^k$ .

Of course, for any permutation  $J^k$  of the indices, it is possible to construct a  $\mathbf{Q}^k$  using Householder reflectors such that  $\mathbf{A}^k$ ,  $\mathbf{B}^k$ , and  $\mathbf{C}^k$  are of the correct form. In [31], it was shown that perfectly stable and highly accurate sensors, in the sense of Theorem 1, may be obtained by maximizing  $|\det \mathbf{A}^k|$  over all possible permutations  $J^k$ . Because the  $\mathbf{A}^k$  are upper triangular, the column-pivoted QR procedure may be seen as a greedy method which approximates the optimal solution iteratively, updating  $J^{k+1}$  so that  $|\det \mathbf{A}^{k+1}|$  is as large as possible with the first  $k$  entries of  $J^{k+1}$  fixed to be equal to the first  $k$  entries of  $J^k$ .

*Remark 1:* For the calculations in Sections V and VI, we computed  $\hat{\mathbf{T}}(J)$  using the formula  $\hat{\mathbf{T}}(J) = \mathbf{X}_{\cdot J}^\dagger \mathbf{X}$  and found that this worked well for our examples, though there may be an advantage to using a more numerically stable definition for  $\hat{\mathbf{T}}(J)$ , see (3.13) of [32].

### III. ALGORITHM FOR SENSOR PLACEMENT UNDER COST CONSTRAINTS

We can now derive a greedy algorithm for the relaxed version of the cost-constrained sensor placement problem (6). The algorithm is based on the column-pivoted QR algorithm described in Section II-C, where the pivot column is now chosen to balance the decrease in the error  $e(J)$  with the increase in the total cost  $\sum_{j \in J} \eta_j$ .

Let  $J^k$ ,  $\mathbf{Q}^k$ ,  $\mathbf{A}^k$ ,  $\mathbf{B}^k$ , and  $\mathbf{C}^k$  define a partial QR factorization at step  $k$ , as in (12). To incorporate the effect of the cost of a sensor, we update the factorization by finding the  $l$  which satisfies

$$l = \arg \max_{i=1, \dots, n-k} \|\mathbf{c}^{k,i}\|_2 - \gamma \eta_{j_{i+k}^k}, \quad (17)$$

where  $\gamma$  is some fixed parameter in the optimization. As before, we can update the permutation indices to move this column to the front, i.e.  $j_{k+1}^{k+1} = j_{k+l}^k$ ,  $j_{k+l}^{k+1} = j_{k+1}^k$ , and  $j_i^{k+1} = j_i^k$  for all other  $i$ , and update the orthogonal matrix  $\mathbf{Q}^k$  via

$$\mathbf{Q}^{k+1} = \mathbf{Q}^k \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{H}(\mathbf{c}^{k,l}) \end{pmatrix}, \quad (18)$$

so that the corresponding  $\mathbf{A}^{k+1}$ ,  $\mathbf{B}^{k+1}$ , and  $\mathbf{C}^{k+1}$  are of the proper form. This process is summarized in pseudocode in

---

#### Algorithm 1 QR pivoting with cost constraints.

---

**Input:** data matrix  $\mathbf{X}$ , number of sensors  $p$ , cost vector  $\boldsymbol{\eta}$ , cost-error balance  $\gamma$

**Output:** partial QR decomposition in  $\mathbf{Q}$  and  $\mathbf{R}$ , and pivots in  $J$  (the first  $p$  are the sensors)

```

1:  $m, n \leftarrow \text{size}(\mathbf{X})$ 
2:  $\mathbf{R} \leftarrow \text{copy}(\mathbf{X})$ 
3:  $\mathbf{Q} \leftarrow \text{eye}(m)$ 
4:  $J \leftarrow 1 : n$ 
5: for  $k = 1, \dots, p$  do
6:   for  $i = k, \dots, n$  do
7:      $\nu_i \leftarrow \|\mathbf{R}_{k:m,i}\|_2 - \gamma \eta_{j_i}$ 
8:    $l \leftarrow$  index of the maximum of  $\nu_{k:n}$   $\triangleright$  choose the
   pivot as in (17)
9:    $\mathbf{v} \leftarrow \mathbf{R}_{k:m,k-1+l}$ 
10:   $\text{swap}(\mathbf{R}_{k:m,k}, \mathbf{R}_{k:m,k-1+l})$ 
11:   $\text{swap}(j_k, j_{k-1+l})$ 
12:   $\sigma \leftarrow \|\mathbf{v}\|_2$ 
13:   $\mathbf{u} \leftarrow (\mathbf{v} + \text{sign}(v_1)\sigma \mathbf{e}^1) / \sqrt{2\sigma(\sigma + |v_1|)}$   $\triangleright$  build the
   normalized Householder reflector
14:   $\mathbf{R}_{k:m,k:n} \leftarrow \mathbf{R}_{k:m,k:n} - 2\mathbf{u}\mathbf{u}^\top \mathbf{R}_{k:m,k:n}$ 
15:   $\mathbf{Q}_{:,k:n} \leftarrow \mathbf{Q}_{:,k:n} - 2\mathbf{Q}_{:,k:n} \mathbf{u}\mathbf{u}^\top$ 

```

---

Algorithm 1. Let  $n$  denote the dimension of the signal,  $m$  denote the number of training samples, and  $p$  denote the desired number of sensors. Then, the algorithm has a computational complexity of  $O(mnp)$ , which can be determined by counting operations in the pseudocode. Note that typically the vectors corresponding to the Householder reflector are stored rather than the matrix  $\mathbf{Q}$  itself in an efficient implementation.

When  $\gamma = 0$ , the pivot chosen by the formula (17) is the same as the pivot in column pivoted QR. After  $k+1$  steps, the difference in the error,  $e(J_{1:k}) - e(J_{1:k+1})$ , is at least  $\|\mathbf{c}^{k,l}\|_2 / (\|\mathbf{X}\|_F (1 + 2\sqrt{n-k}))$ . Therefore, a positive  $\gamma$  balances the decrease in the error,  $e(J_{1:i}) - e(J_{1:i+1})$ , with the cost of the pivot,  $\eta_{j_{i+1}}$ .

*Remark 2:* The pivot as chosen in (17) does not necessarily correspond to the natural greedy choice, i.e. the pivot which minimizes  $e(J_{1:k+1}) + \gamma \sum_{j \in J_{1:k+1}} \eta_j$  with  $J_{1:k}$  fixed. Such a pivot could be computed, though at greater cost than the present algorithm. Further, there is another reason to avoid such a strategy: it completely ignores the stability of the resulting map. By instead pivoting based on column size, we bias the algorithm toward choosing stable pivots while still incorporating some sense of the reduction in error.

### IV. DESCRIPTION OF DATA SETS

In this section, we provide details about the three example data sets we consider, as well as the cost functions and parameters used. An example snapshot from each system is given in Figure 1.

We will consider two types of training set: interpolative and extrapolative. By interpolative, we mean that we have sampled a subset of the data that draws from all regions of the parameter space. By extrapolative, we mean that we have purposefully missed data from a portion of the parameter space.

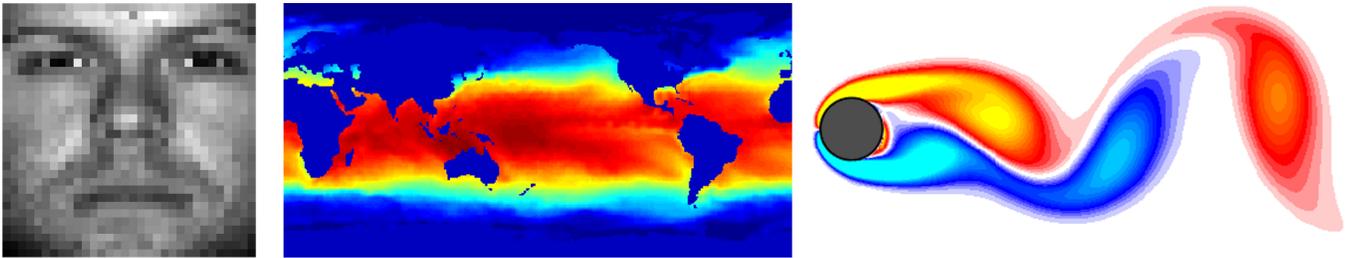


Fig. 1. An example from each of the three data sets considered in this work. Pictured from left to right are one of the Yale B faces, a snapshot of global sea surface temperature, and a snapshot of vortex shedding of a fluid flowing around a stationary cylinder.

The data that is left out forms the testing set. Finding good sensor locations for the extrapolative training sets is a harder problem; the sensor locations must reasonably generalize to samples of data which may be unlike anything in the training set.

The first example is the Extended Yale Face Database B, referred to here as the eigenface data set [49]–[52]. It comprises about 64 images each of 38 individuals under various lighting conditions. The images are downsized to  $32 \times 32$  pixels. Unless otherwise stated, the tests are conducted on an interpolative training set, by randomly selecting 80% of the images. A 20-fold cross validation is performed. When an extrapolative training set is used, the algorithm is trained on the first 80% of the images, meaning that some individuals appear only in the test set. Three cost functions are tested for the eigenfaces: (i) a Gaussian function, such that it is most expensive to place sensors in the center of the face, (ii) a step function uniformly penalizing sensors in the middle ninth of the photographs, and (iii) a step function penalizing the left third of the images. All three cost functions are plotted in the left column of Figure 4. The range of values used for the parameter  $\gamma$  is dependent on the cost function considered. With the Gaussian cost function,  $\gamma$  ranges from 0 to  $1.9 \times 10^5$ , while the two step functions both take  $\gamma$  between 0 and  $6 \times 10^4$ .

The next data set we consider is the NOAA\_OISST\_V2 mean sea surface temperature set [53]–[55], comprising weekly global sea surface temperature measurements between the years of 1990 and 2016. There are a total of 1400 snapshots on a  $360 \times 180$  spatial grid. The algorithm is trained on 1100 snapshots (randomly selected for the interpolative training set, or the first 1100 in sequence for the extrapolative set) and tested on the remaining 300. Ten cross validations are performed for the interpolative trials. The cost function used is a step function that penalizes placing sensors too far from shore, being zero for locations one and two pixels off land, and equal to one everywhere else. Values of  $\gamma$  range from 0 to 225.

Our final example is the vortex shedding of a fluid flowing around a stationary cylinder. The flow data is generated using the immersed boundary projection method [56], [57] to numerically simulate the Navier-Stokes equations with Reynolds number 100. There are 151 snapshots in time, each on a  $199 \times 449$  spatial grid. The training sets consist of 120 snapshots. As before, these snapshots are chosen randomly for the interpolative training set, while the first 120 are taken

in sequence to form the extrapolative training set. In the interpolative case, 30 cross validations are performed. We choose a cost function that is uniformly equal to one in the lower half of the domain and zero in the upper half, to exploit the symmetry of the system. Values of  $\gamma$  are between 0 and 15.

These data sets were chosen because they represent classic examples from several branches of the sensor placement literature. The face data has been used to demonstrate classification methods by discriminating between men and women, or between individuals, as in [2], [24], [58]. Sensor placement for ocean data reconstruction has a long history, including the papers [4]–[6], in which sensors were placed at the extrema of the POD modes. The flow behind a stationary cylinder is a prototypical example from the reduced order modeling community, which often uses the empirical interpolation method [10] or the discrete empirical interpolation method (DEIM) [11], [12] to create a low-rank model of a system for future state prediction. See [9] and the citations therein for a more in-depth exploration of the data sets and how they have been analyzed historically. Some of the existing methods, such as gappy POD and DEIM, could be adapted to include a heterogeneous sensor cost function; however, we have chosen to adapt the pivoted QR algorithm because it is robust, computationally efficient, and when unmodified it provides near-optimal results [9], [12], [32]. Many of the existing methods for these problems perform some pre-processing, often by applying the algorithm to singular vectors (or POD modes) of the data rather than the data directly, a concept which we will explore in the next section.

*Remark 3:* The weighting  $\gamma$  is an arbitrary scalar, and its values were chosen by trial and error to ensure that a cost-error curve was traced out in sufficient detail for each data set and cost function.

## V. DATA, SINGULAR VECTORS, AND RANDOM PROJECTIONS

Before proceeding to the cost-constrained placement examples, we will first briefly discuss the question of data pre-processing for sensor placement. In the notation of Section II, pre-processing refers to the process of creating the matrix  $\Psi^{tr}$  from the training data  $\mathbf{X}^{tr}$  (we then apply the QR-based algorithm to  $\Psi^{tr}$ ). When selecting  $p$  sensors, a common choice for the matrix  $\Psi^{tr}$  is to set it as the first  $p$  right singular vectors of  $\mathbf{X}^{tr}$  [9], [11], [12]. Inspired by the randomized

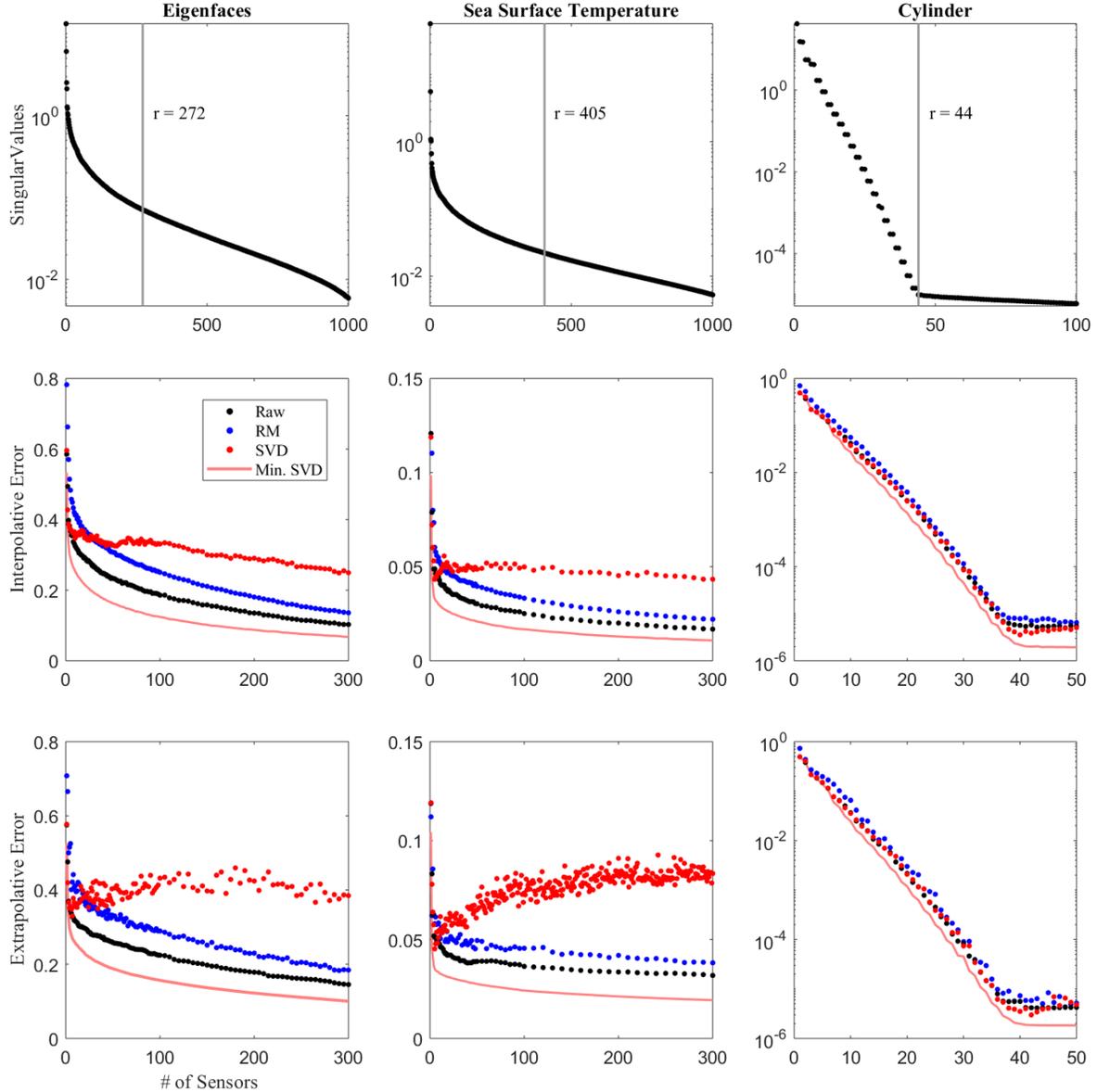


Fig. 2. Reconstruction error versus the number of sensors for the three data sets described in Section IV, using  $p$  SVD modes,  $2p$  randomized linear combinations (abbreviated RM in the legend, for randomized modes), and the raw data without pre-processing. The top row shows a log plot of the normalized singular values, with the vertical gray line indicating the Gavish-Donoho cutoff  $r$  [59]. The remaining plots show the average reconstruction error given sensors placed by performing the unmodified column-pivoted QR decomposition on  $\Psi^{tr}$  formed using the three pre-processing methods discussed in the text. The first column provides eigenface results, the second gives sea surface temperature reconstruction errors, and the third shows errors for the flow behind a cylinder on a log scale. The middle row of the figure shows interpolative error, where the training set consists of a randomly-chosen subset of the data, while the bottom row gives the extrapolative data, which takes the first 80% of the parameter space. All plots also show a rough estimate of the minimum error at a given number of sensors (the solid red line), obtained by projecting the full image onto the SVD modes.

linear algebra community [35], [36], we also consider setting  $\Psi^{tr} = \mathbf{G}\mathbf{X}^{tr}$ , where the entries of  $\mathbf{G} \in \mathbb{R}^{2p \times m}$  are drawn from a standard normal distribution, i.e. we set the rows of  $\Psi^{tr}$  to be random linear combinations of the rows of  $\mathbf{X}^{tr}$ . We also consider the performance when setting  $\Psi^{tr} = \mathbf{X}^{tr}$ , i.e. the performance without pre-processing. The number of singular vectors or random linear combinations used is open-ended; we found that, as described above,  $p$  singular vectors and  $2p$  random linear combinations gave reasonably optimal performances for each pre-processing technique, respectively.

Our data sets are the Extended Yale Face Database B, the

Optimally Interpolated Sea Surface Temperature data set from NOAA, and simulation data for fluid flow behind a cylinder. For the face and sea surface temperature data, we consider both interpolative and extrapolative training sets. In this section, we do not make such a distinction for the fluid simulation data, as it is much lower rank and periodic in time.

The top row of Figure 2 shows the spectrum of normalized singular values for all three data sets. These plots include a gray line at the Gavish-Donoho [59] optimal hard-threshold cutoff, which is an estimate of the rank beyond which the SVD modes represent additive noise. It is apparent that the

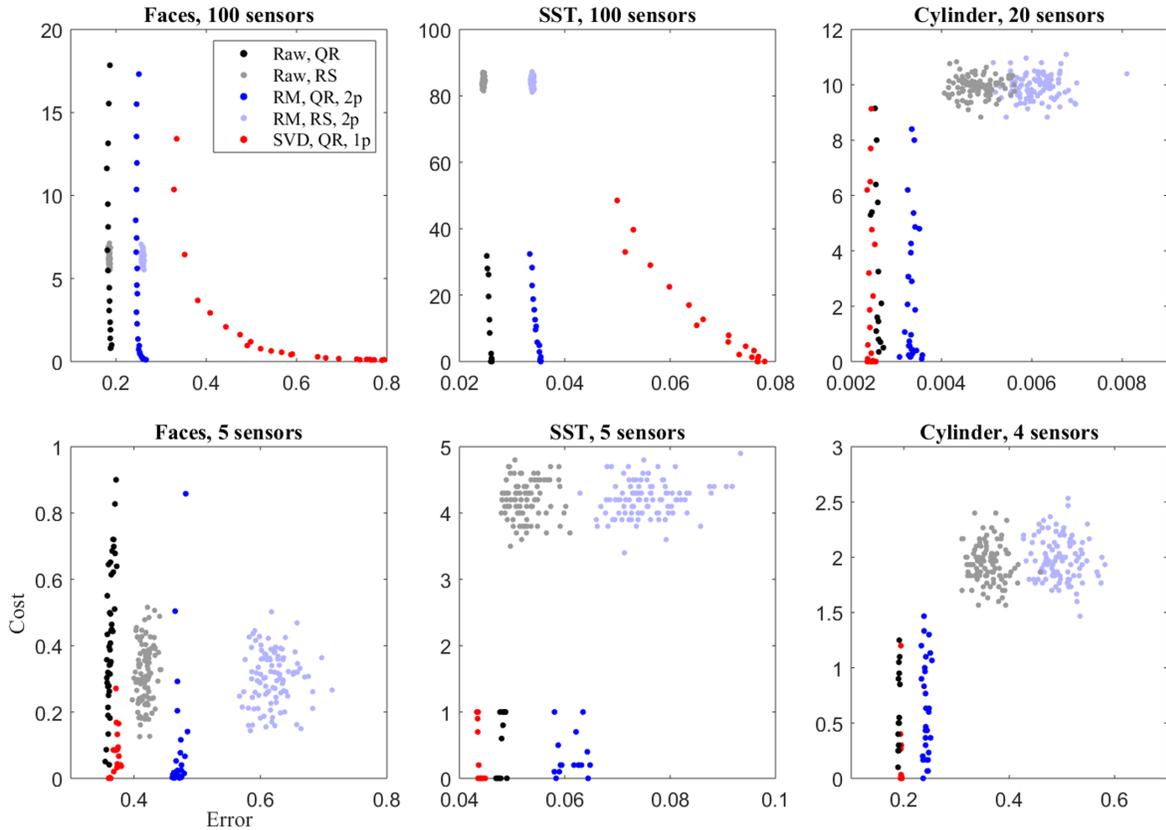


Fig. 3. A comparison of cost versus error results with the following pre-processing methods: Taking the full raw data as a basis,  $\Psi^{tr} = \mathbf{X}^{tr}$ ; taking  $2p$  randomized linear combinations of  $\mathbf{X}^{tr}$ , where  $p$  is the number of sensors; and taking  $\Psi^{tr}$  to be the first  $p$  SVD modes of  $\mathbf{X}^{tr}$ . The training set is interpolative, not extrapolative. Sensors are placed with our modified QR-based algorithm for each of the three bases and data sets, for both a comparatively large and small number of sensors. Additionally, cost and error results are calculated for randomly-selected sensors with the first two pre-processing methods. The randomized sensor results are cross validated as usual, and there are 100 data points per method. Randomized sensors using the SVD basis had significantly higher reconstruction errors than any other method, and the results are not shown here. A Gaussian cost function is used for the eigenface example. The abbreviations in the legend are explained in Table I.

fluid flow data set is fundamentally different from the other two, having a sharp elbow at the cutoff, as opposed to a slow decay.

The remainder of Figure 2 plots the relative interpolation error (4) computed for the test set as a function of the number of sensors using each of the three methods for pre-processing (SVD modes, randomized modes, and the raw data) described above. The sea surface temperature and eigenface data sets both have an interesting feature not present in the fluid flow data: the error from the SVD basis has a local minimum at a very small number of modes — five for the temperature data and approximately seventy for the eigenfaces. While the error for the interpolative training set begins to slowly decrease again as the number of sensors is increased, the performance of the extrapolative data gets worse as more sensors are added, up until at least 150 sensors. This unexpected behavior reveals that, for systems with slow singular value decay, there is an overfitting problem when using SVD modes which occurs well before those modes correspond to additive sensor noise. We believe that the source of this problem is in the formation of the interpolant  $\hat{\mathbf{T}}(J)$  via the formula  $\hat{\mathbf{T}}(J) = \Psi_{\cdot J}^{tr\dagger} \Psi^{tr}$ . When the rows of  $\Psi^{tr}$  consist of right singular vectors of  $\mathbf{X}^{tr}$  — which generally correspond to smooth, large scale structures

— some of the small-scale variability of the data is lost and the interpolant overfits to these smoother modes. The problem is that, with slow singular value decay, an additional singular vector may not resolve much more fine structure than the previous.

The reconstruction errors for sensors based on random linear combinations or the raw data do not have this behavior (except for a weak effect with extrapolative sea surface temperature data), nor do any of the cylinder trials. Indeed, the error for the random linear combinations and the raw data behaves as expected, decaying at a rate that follows the error obtained from projecting the test set onto the first  $p$  singular modes of the training set (this rough indicator of optimal behavior is plotted as a solid red line in the figure).

We make a few conclusions based on these pre-processing results. If the goal of pre-processing is to improve the quality of the sensors, then it appears that using no pre-processing is the preferred method, except when placing a very small number of sensors, where the SVD mode method displays an advantage. If the goal of the pre-processing is to reduce the size of the problem and speed up the optimization procedure, then it appears that using randomized linear combinations of the modes is preferable to using a limited number of SVD

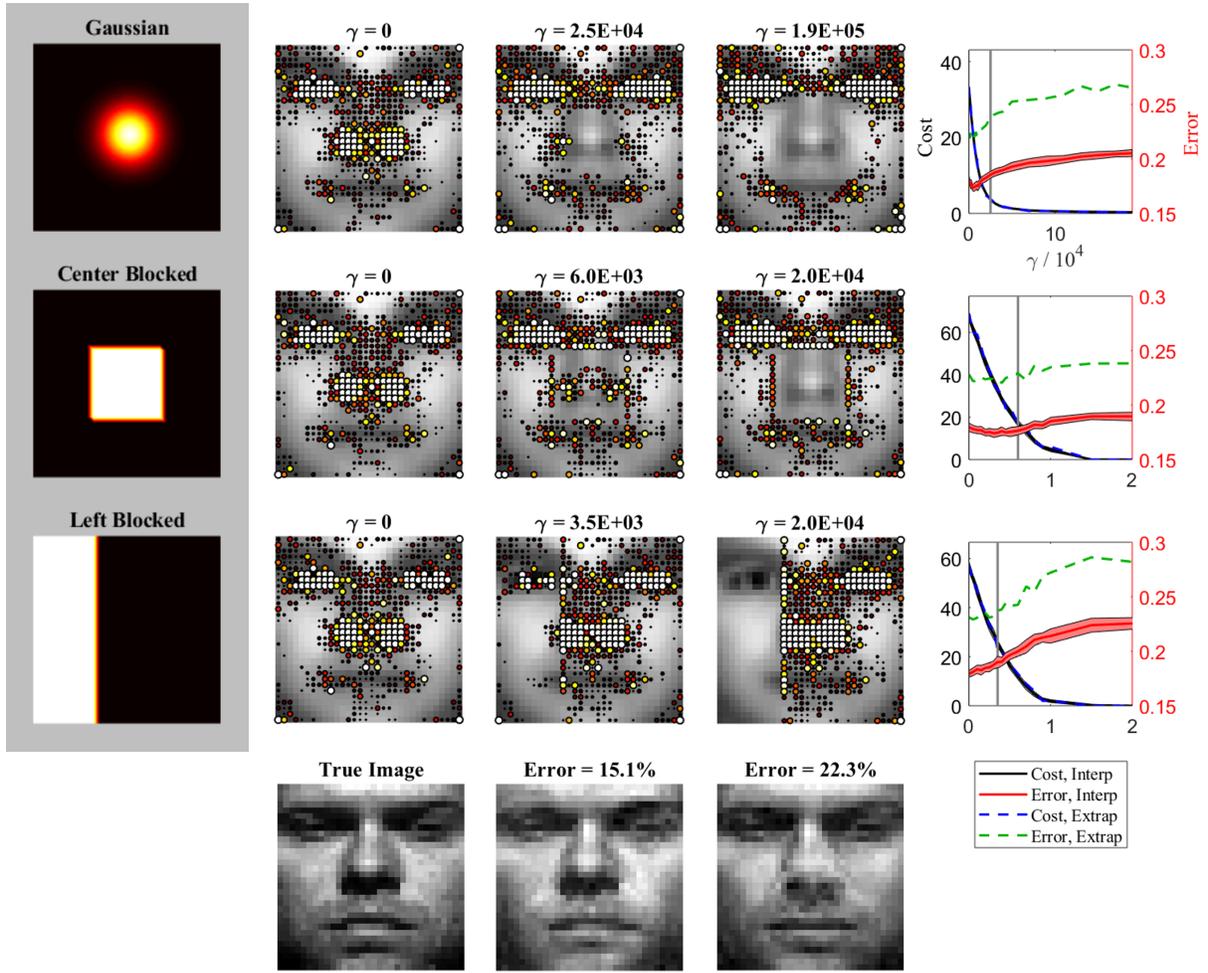


Fig. 4. Average sensor locations for eigenface reconstruction for three different cost functions. The cost functions are plotted in the left column, where white indicates regions of highest cost and black shows regions of zero cost. The three central columns show the locations of 200 sensors placed by the QR-based algorithm, averaged over 20 cross validations, for increasing values of the weighting factor  $\gamma$ . The final column plots cost and error against  $\gamma$  for each cost function, with bands indicating error bars. Both interpolation and extrapolation results are given. The vertical gray lines indicate the value of  $\gamma$  at which the sensors are plotted in the middle column. The bottom row shows typical example reconstructions of one of the photos, for reference.

modes. (Note that these SVD modes would have to be computed with an accelerated procedure in order for using them to represent a speed-up for the QR-based sensor placement algorithm, again, with the caveat that SVD modes behave better for a small number of sensors.)

## VI. APPLICATIONS

When factoring in the effect of cost, we observe behavior similar to the cost-free case analyzed in the previous section, with respect to pre-processing. Figure 3 provides an overview of the performance for the various pre-processing methods, now with a non-zero cost associated with each location. In the plots, we generate several cost, error pairs for each pre-processing technique by varying the cost function weighting  $\gamma$  in the QR-based algorithm. We also plot the performance of randomly drawn sensors for the sake of comparison. All three data sets are considered, with both a large and small number of sensors. See Table I for details on the figure labels.

For both the eigenface and sea surface temperature data sets, using the raw data at 100 sensors leads to the lowest error

Color	Method	Description
●	Raw, QR	Performing the QR-based algorithm directly on the raw data.
●	Raw, RS	Randomly-placed sensors, using the raw data as a basis.
●	RM, QR, $2p$	The QR-based algorithm on the randomized modes $\Psi^{tr} = \mathbf{G}\mathbf{X}^{tr}$ , $\mathbf{G} \in \mathbb{R}^{2p \times m}$ a matrix with randomized entries.
●	RM, RS, $2p$	Using randomly-selected sensor locations, and $2p$ randomized modes as a basis.
●	SVD, QR, $1p$	The QR-based algorithm on the first $p$ SVD modes.

TABLE I  
A BRIEF DESCRIPTION OF THE SENSOR PLACEMENT METHODS USED TO CREATE FIGURE 3, AND THE COLORS IN WHICH THEY ARE PLOTTED.

at a comparable cost, and randomized linear combinations with  $2p$  modes gives the next lowest error, followed by SVD with  $p$  modes. At 5 sensors, the latter is reversed, with SVD performing comparably to or better than the raw data, as is

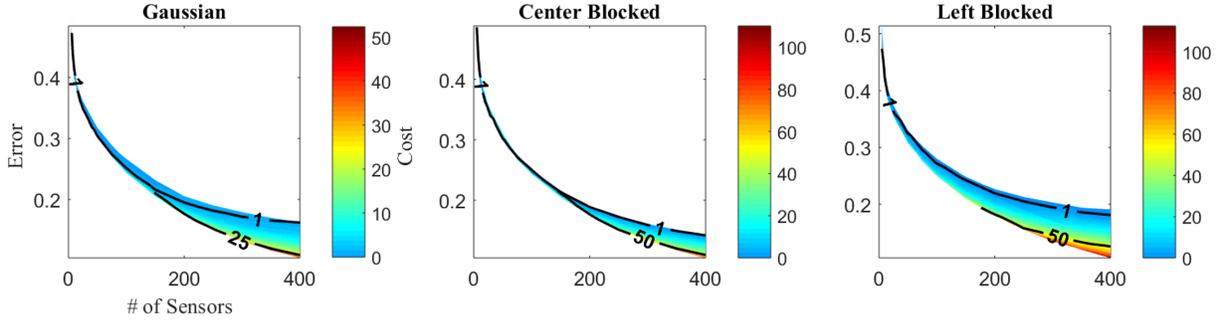


Fig. 5. The cost landscape for eigenface reconstruction for all three cost functions, with cost plotted as a two-dimensional color map against error and the number of sensors. Contours indicate lines of constant cost.

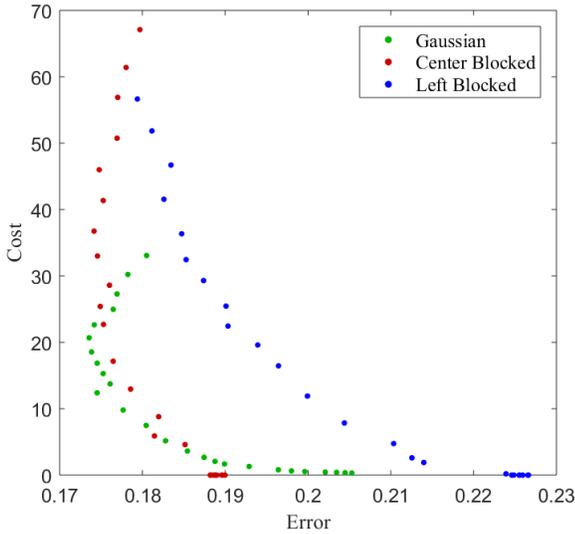


Fig. 6. Cost versus error for eigenface reconstruction with three different cost functions. Results are given for the case of 200 sensors.

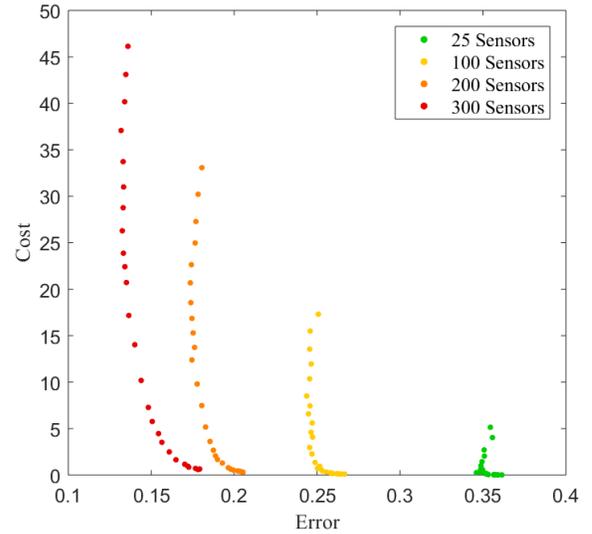


Fig. 7. Cost versus error for eigenface reconstruction with a Gaussian cost function, for several different numbers of sensors.

the case with both trials for the fluid flow behind a cylinder. We observe that our QR-based method generally outperforms the best randomly-chosen sensors. The randomized sensors sometimes yield lower errors, particularly with a large number of sensors, but our principled method is capable of producing lower costs. We note that the randomized data performs worse than the raw data in all examples, but the behavior is consistent and may be worth the reduced computational cost in some applications.

In the remainder of this section, we more closely analyze the performance of our algorithm for each of the data sets. For brevity, we consistently use a randomized linear combination of data vectors for the pre-processing technique in these examples, noting that the behavior described above would be maintained if all techniques were tested.

#### A. Eigenfaces

The algorithms are first tested on the eigenface data set. All three cost functions and a few corresponding sensor arrays are shown in Figure 4. These are the average locations from placing 200 sensors over the twenty cross-validation runs, shown as a scatter plot on top of the dominant eigenface mode. Marker size and color indicate the frequency with which a

sensor was placed at a given location, with white indicating the most frequent, shading through yellow and red to black for the least frequent. As expected, when the cost function weighting is increased, sensors are gradually pushed out of the regions of higher cost. This allows the total cost to be lowered at the expense of decreasing reconstruction accuracy, as demonstrated in the right-hand column of the figure, which plots cost and error on separate axes, as a function of  $\gamma$ . Note that  $\gamma$  is an arbitrary weighting, and the same value of  $\gamma$  can have very different effects depending on the cost function. Extrapolative cost and error are shown in the same panel, where the extrapolative error is higher than the interpolative, at an identical cost. Additionally, the bottom row of the figure shows several reconstruction examples for one of the faces.

For many practical applications of these methods, the final goal will be to minimize reconstruction error given a predetermined budget. To that end, cost-error landscapes are constructed by calculating sensor array cost and reconstruction error for different numbers of sensors. The results are shown in Figure 5, which shows the landscapes for each cost function, plotted as a color map according to cost. Cost contours on this color map represent a hypothetical budget, so the optimum configuration for a given budget can be found by following the

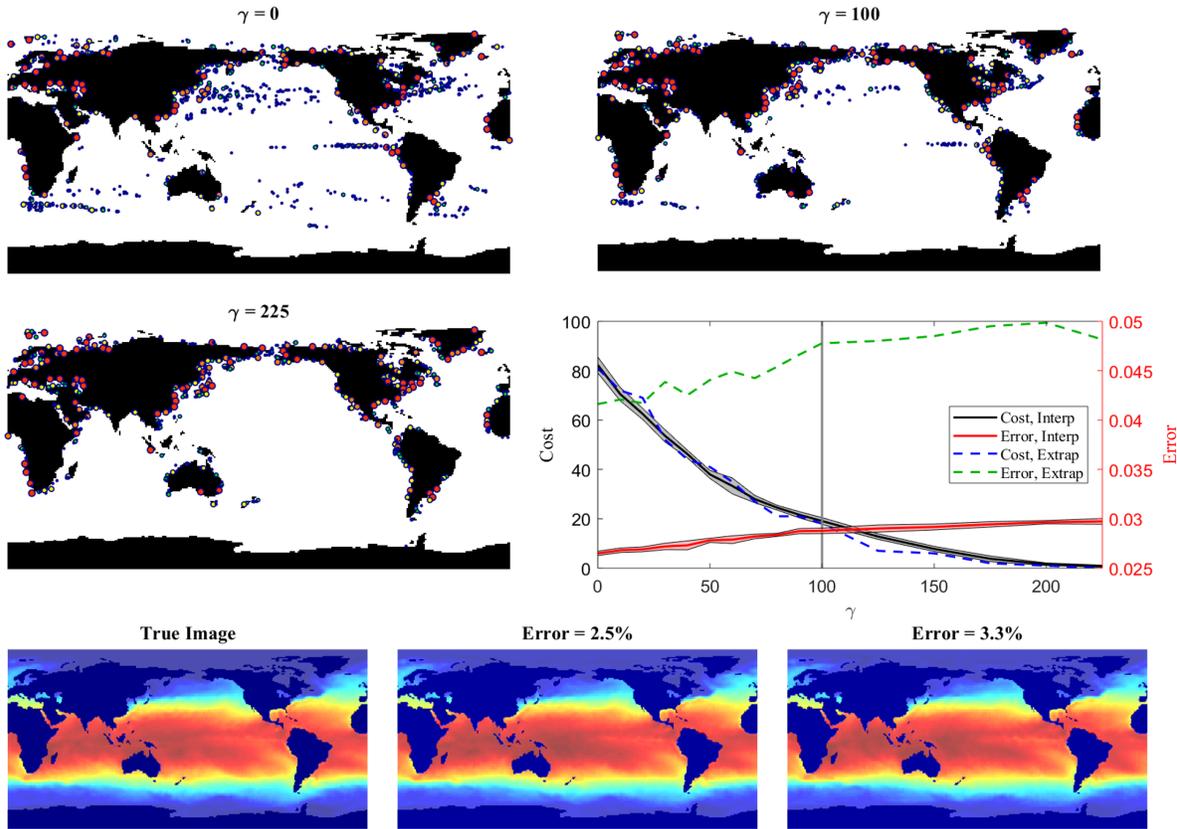


Fig. 8. Sensor locations for sea surface temperature reconstruction with 200 sensors. The cost function considered was a step function which was zero up to two pixels off land and equal to one everywhere else. Locations are shown for three different values of the weighting factor  $\gamma$ , and are averaged over ten cross validations. Size and color of a data point indicate the frequency with which a sensor was placed there. The fourth image plots cost and error against  $\gamma$ , for both interpolative and extrapolative data sets. The bottom row shows a comparison of an example temperature snapshot along with two reconstructions of it yielding two different accuracies.

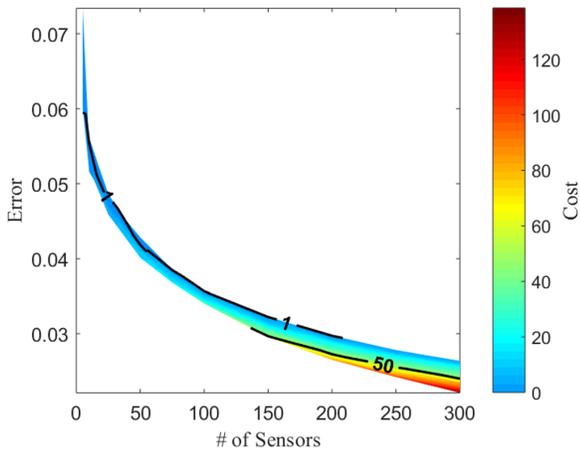


Fig. 9. The cost landscape for sea surface temperature reconstruction, plotted as a color map against error and the number of sensors. Contours show lines of constant cost.

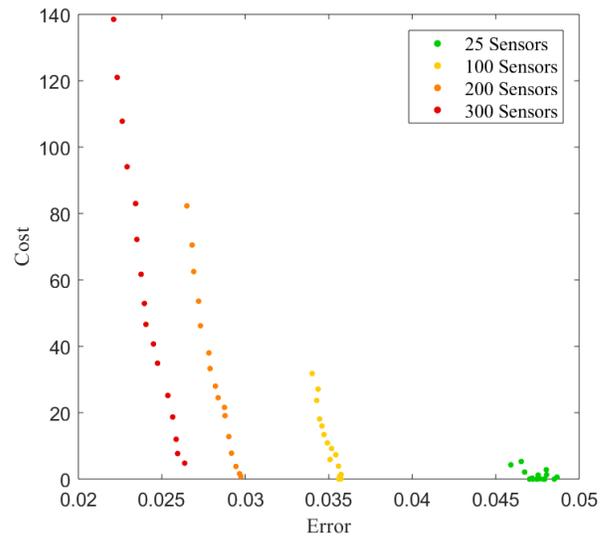


Fig. 10. Plots of cost versus error for sea surface temperature reconstruction, with varying numbers of sensors.

appropriate contour to the lowest possible error. Note that the upper edges of the contour plots indicate the minimum cost and maximum error for a given number of sensors, and the

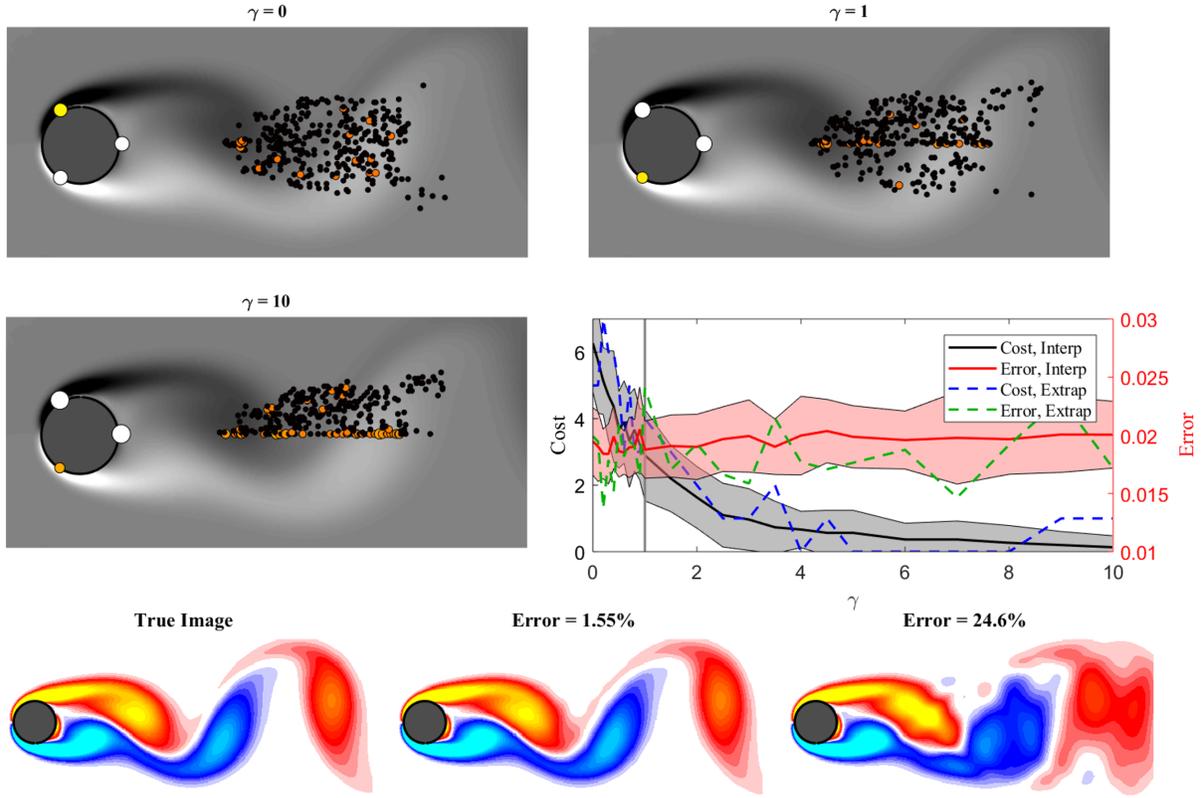


Fig. 11. Sensor locations and reconstructions of the flow behind a cylinder, obtained by the QR-based algorithm, using 14 sensors. The cost function was a step function blocking the lower half of the domain. The first three plots show the placements of the sensors averaged over 30 trials. The fourth shows cost and error plotted against  $\gamma$ , where the bands indicate error bars. The bottom row gives a comparison of the true image of a snapshot along with two different reconstructions.

lower edges indicate the minimum error and maximum cost.

Cross sections of the cost landscapes as plots of cost versus error are given in Figures 6 and 7. The former shows cross sections for each of the three cost functions, using 200 sensors, and the latter shows cost versus error for a Gaussian cost function with 25, 100, 200, and 300 sensors.

### B. Sea surface temperature

Next we consider the sea surface temperature data set. Average sensor locations over ten cross validations with 200 sensors are shown in Figure 8, as a scatter plot where the size and color of a data point indicate the frequency with which a sensor was placed in that location (blue being the least frequent, red being the most frequent). As the cost function weighting is increased, more sensors move within the unblocked regions offshore, until the cost reaches zero. Plots of cost and error as functions of  $\gamma$  are given in the fourth panel, and the bottom row shows two example reconstructions. As with the eigenfaces, the interpolative trial has a much lower error than the extrapolative trial. Notice that although the reconstruction error increases as cost decreases, here the error does not even reach 3%, even when the cost is zero.

The landscape of cost as a function of error and the number of sensors is shown in Figure 9, again visualized as a color map with contours of constant cost. Cross sections of cost

versus error for 25, 100, 200, and 300 sensors are shown in Figure 10.

### C. Fluid flow around a cylinder

Finally, we test the algorithm on the fluid flowing around a stationary cylinder. This data set is low-rank, periodic, and vertically symmetric, making it a significant contrast to the previous two examples. The vertically symmetric cost function further allows the algorithm to take full advantage of the symmetry of the fluid flow. In Fig. 11, sensor locations for several values of  $\gamma$  are shown. The locations are shown for 14 sensors, averaged over the 30 cross validations and then graphed as a scatter plot on top of an example fluid flow snapshot. The size and color of a data point indicate how frequently a sensor was placed at its location, with white being the most frequent and black being the least frequent. As expected, when  $\gamma$  is increased, the sensors migrate until they occupy the upper half of the plane.

The fourth panel of the figure plots the cost and error on separate axes, as functions of  $\gamma$ . Because of the symmetry of the data set, the reconstruction error is essentially unchanged with  $\gamma$ , even as the cost goes to zero. Furthermore, because the flow is periodic, the extrapolative data performs slightly better than the interpolative data. The figure's bottom row shows example reconstructions of a snapshot.

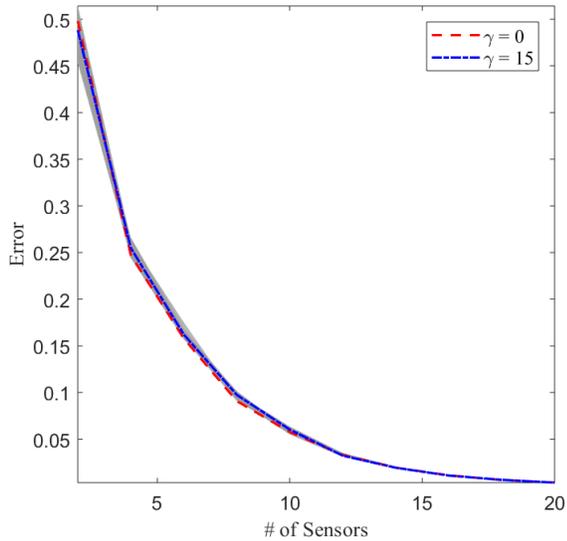


Fig. 12. The cost landscape for reconstruction of a fluid flow behind a cylinder. Instead of the color maps made for the previous two data sets, curves of error versus the number of sensors are shown for high and low values of  $\gamma$ . The gray band indicates the maximum variation in the error.

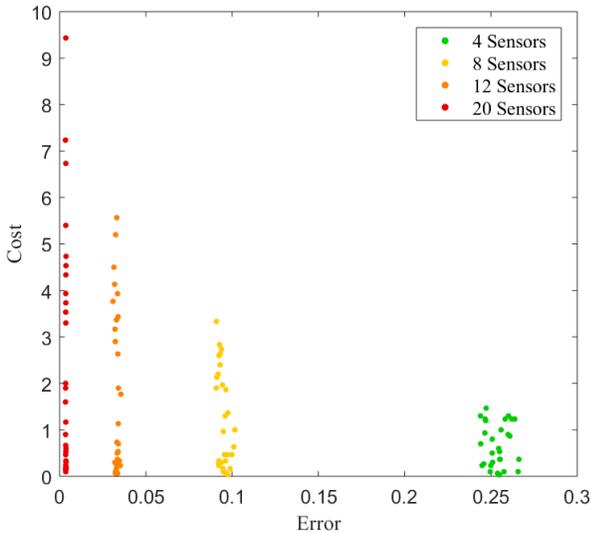


Fig. 13. Plots of cost versus error for cylinder flow reconstruction, with varying numbers of sensors.

The fluid flow cost-error landscape is shown in Fig. 12. Notice that there is hardly any variation in the error, so rather than plotting cost as a color map, the error versus number of sensor curves for  $\gamma = 0$  (highest cost) and  $\gamma = 15$  (lowest cost) are shown. These curves are essentially identical, further emphasizing that for this particular data set and cost function, the total cost can be lowered with no penalty to the error. Cost versus error plots are given in Fig. 13, which shows results for 4, 8, 12, and 20 sensors. Regardless of the number of sensors, these cross sections are essentially vertical lines, within the standard deviation of the error.

Data set	$m$	$n$	$p$	$t$
Yale Faces	1931	1024	400	0.82
Cylinder	129	89351	20	0.52
Sea Surface Temperature	1100	44219	300	25.65

TABLE II  
AVERAGE RUN TIME INFORMATION FOR RUNNING ALGORITHM 1 ON THE RAW DATA FOR EACH APPLICATION.

#### D. Run time for the experiments

To get a sense of how long it takes to compute sensor locations, we provide some average run time information in Table II. Note that  $m$  denotes the number of training samples,  $n$  denotes the dimension of the data,  $p$  denotes the number of sensors, and  $t$  denotes the average run time in seconds<sup>2</sup> for a single value of  $\gamma$ . For all applications,  $p$  is selected to be the maximum number of sensors placed in any of the experiments above. The times are averaged over 26 runs for the Yale faces data, 5 runs for the sea surface temperature data, and 29 runs for the cylinder flow data. All times are for the raw data without preprocessing.

## VII. CONCLUSION AND FUTURE DIRECTIONS

We have developed a QR-based greedy algorithm to place sensors for reconstruction with a cost constraint on sensor locations. This algorithm is tested on three different data sets, eigenfaces, weekly sea surface temperature data, and vortex shedding of a fluid flowing around a cylinder. In all cases, the method is demonstrated to be capable of lowering sensor cost at the expense of marginal increases in reconstruction error.

It is also shown that with or without the inclusion of a cost function, data sets with slow singular value decay have better results by pre-processing the data through a randomized linear combination of modes, rather than through SVD-based rank reduction. Randomized linear combinations lead to significantly lower reconstruction errors, except at a very low number of sensors.

In fact, for these data sets with slow singular value decay, SVD modes behave in an unexpected way at a low number of modes. The reconstruction error decreases sharply, even surpassing the error obtained by using the full raw data in the case of sea surface temperatures, before increasing again as more sensors are added. This suggests that the SVD is overfitting well before the Gavish-Donoho cutoff, an idea which warrants further exploration in future work. The results also imply that there may be some other pre-processing method which can take advantage of both the SVD behavior at a low number of sensors and the random linear combination or raw data behavior at a higher number of sensors. This will also be explored in the future.

In the meantime, the algorithm presented here provides a way to place sensors under a cost constraint, which could have applications in manufacturing, atmospheric sensing, fluid flow sensing, and many more fields. Specifically, the algorithm allows one to address three critical engineering design principles in regard to sensors placement: (i) For a fixed budget

<sup>2</sup>All experiments were run on a laptop with an Intel Core i7-6600u CPU (2.60 GHz, 4 cores) and 16 Gb of RAM.

of sensors, where are the best measurement locations, (ii) What is the minimal number of sensors required to achieve a given reconstruction error, and (iii) How well can inaccessible regions be reconstructed in practice. Depending upon the application, one or all of these questions may be of central concern. The computationally tractable approach presented here provides a principled mathematical method for answering these questions.

One weakness of the approach advocated here is the lack of performance guarantees. We believe that it is worth exploring in a systematic manner both the effect of using a more cost-intensive selection scheme, as in the Gu and Eisenstat approach [31], and the effect of using performance metrics other than the determinant, like the frame potential [18] and mutual information [39], [40], in the cost-constrained reconstruction-oriented setting. This will be the subject of future research.

#### ACKNOWLEDGMENT

T. Askham would like to thank Mark Tygert for useful discussions of the interpolative decomposition. T. Askham and J. N. Kutz acknowledge support from the Air Force Office of Scientific Research (FA9550-15-1-0385). S. L. Brunton acknowledges support from the Air Force Research Laboratory (FA8651-16-1-0003) and the Air Force Office of Scientific Research through the Young Investigator Program (FA9550-18-1-0200).

#### REFERENCES

- [1] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse sensor placement optimization for classification," *SIAM Journal on Applied Mathematics*, vol. 76, no. 5, pp. 2099–2122, 2016.
- [2] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.
- [3] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.
- [4] Y. Zhang and J. G. Bellingham, "An efficient method of selecting ocean observing locations for capturing the leading modes and reconstructing the full field," *Journal of Geophysical Research: Oceans*, vol. 113, no. C4, 2008.
- [5] B. Yildirim, C. Chryssostomidis, and G. Karniadakis, "Efficient sensor placement for ocean measurements using low-dimensional concepts," *Ocean Modelling*, vol. 27, no. 3–4, pp. 160–173, 2009.
- [6] X. Yang, D. Venturi, C. Chen, C. Chryssostomidis, and G. E. Karniadakis, "EOF-based constrained sensor placement and field reconstruction from noisy ocean measurements: Application to Nantucket Sound," *Journal of Geophysical Research: Oceans*, vol. 115, no. C12, 2010.
- [7] R. Everson and L. Sirovich, "Karhunen–Loève procedure for gappy data," *JOSA A*, vol. 12, no. 8, pp. 1657–1664, 1995.
- [8] K. Willcox, "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Computers & fluids*, vol. 35, no. 2, pp. 208–226, 2006.
- [9] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Systems*, vol. 38, no. 3, pp. 63–86, 2018.
- [10] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathématique*, vol. 339, no. 9, pp. 667–672, 2004.
- [11] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.
- [12] Z. Drmac and S. Gugercin, "A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions," *SIAM Journal on Scientific Computing*, vol. 38, no. 2, pp. A631–A648, 2016.
- [13] T. H. Summers and J. Lygeros, "Optimal sensor and actuator placement in complex dynamical networks," *arXiv preprint arXiv:1306.2491*, 2013.
- [14] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Control and sensing co-design," *arXiv preprint arXiv:1802.08376*, 2018.
- [15] A. Krause and D. Golovin, "Submodular function maximization." 2014.
- [16] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006, pp. 2–10.
- [17] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, 2007, pp. 1650–1654.
- [18] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on signal processing*, vol. 62, no. 5, pp. 1135–1146, 2014.
- [19] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 2572–2577.
- [20] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *arXiv preprint arXiv:1102.3975*, 2011.
- [21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 420–429.
- [22] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," *arXiv preprint arXiv:1207.0560*, 2012.
- [23] P. Businger and G. H. Golub, "Linear least squares solutions by Householder transformations," *Numerische Mathematik*, vol. 7, no. 3, pp. 269–276, 1965.
- [24] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [25] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [26] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [27] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [28] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [29] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.
- [30] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [31] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.
- [32] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin, "On the compression of low rank matrices," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1389–1404, 2005.
- [33] C. Li, S. Jegelka, and S. Sra, "Polynomial time algorithms for dual volume sampling," in *Advances in Neural Information Processing Systems*, 2017, pp. 5038–5047.
- [34] P.-G. Martinsson, V. Rokhlin, and M. Tygert, "On interpolation and integration in finite-dimensional spaces of bounded functions," *Communications in Applied Mathematics and Computational Science*, vol. 1, no. 1, pp. 133–142, 2007.
- [35] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices," *Proceedings of the National Academy of Sciences*, vol. 104, no. 51, pp. 20 167–20 172, 2007.
- [36] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.
- [37] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [38] M. Coutino, S. Chepuri, and G. Leus, "Subset selection for kernel-based signal reconstruction," 2018.

- [39] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 265–272.
- [40] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.
- [41] D. C. Kammer, "Sensor placement for on-orbit modal identification and correlation of large space structures," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 2, pp. 251–259, 1991.
- [42] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [43] K. Lim, "Method for optimal actuator and sensor placement for large flexible structures," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 1, pp. 49–57, 1992.
- [44] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, no. 3, pp. 347–355, 2008.
- [45] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. Control of Network Systems*, vol. 3, no. 1, pp. 91–101, 2016.
- [46] F. L. Cortesi, T. H. Summers, and J. Lygeros, "Submodularity of energy related controllability metrics," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2883–2888.
- [47] I. Shames and T. H. Summers, "Rigid network design via submodular set function optimization," *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 3, pp. 84–96, 2015.
- [48] A. S. Householder, "Unitary triangularization of a nonsymmetric matrix," *Journal of the ACM (JACM)*, vol. 5, no. 4, pp. 339–342, 1958.
- [49] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007.
- [50] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007.
- [51] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal Laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.
- [52] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [53] "NOAA optimal interpolation (OI) sea surface temperature (SST) v2." [Online]. Available: <https://www.esrl.noaa.gov/psd/>
- [54] V. Banzon, T. M. Smith, T. M. Chin, C. Liu, and W. Hankins, "A long-term record of blended satellite and in situ sea-surface temperature for climate monitoring, modeling and environmental studies," *Earth System Science Data*, vol. 8, no. 1, pp. 165–176, 2016. [Online]. Available: <http://www.earth-syst-sci-data.net/8/165/2016/>
- [55] R. W. Reynolds, T. M. Smith, C. Liu, D. B. Chelton, K. S. Casey, and M. G. Schlax, "Daily high-resolution-blended analyses for sea surface temperature," *Journal of Climate*, vol. 20, no. 22, pp. 5473–5496, 2007.
- [56] K. Taira and T. Colonius, "The immersed boundary method: a projection approach," *Journal of Computational Physics*, vol. 225, no. 2, pp. 2118–2137, 2007.
- [57] T. Colonius and K. Taira, "A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, pp. 2131–2146, 2008.
- [58] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Optimal sensor placement and enhanced sparsity for classification," *arXiv preprint arXiv:1310.4217*, 2013.
- [59] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is  $4/\sqrt{3}$ ," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014.



**Emily Clark** received the B.S. degree in physics from Bates College, Lewiston, ME, in 2015. She is a Ph.D. student of physics at the University of Washington.



**Travis Askham** received the B.S. degree in applied mathematics and the M.A. degree in mathematics from the University of California, Los Angeles, Los Angeles, CA, in 2010 and the Ph.D. degree in mathematics from the Courant Institute of Mathematical Sciences at New York University, New York, NY, in 2016. He is a Research Associate of applied mathematics at the University of Washington.



**Steven L. Brunton** received the B.S. degree in mathematics with a minor in control and dynamical systems from the California Institute of Technology, Pasadena, CA, in 2006, and the Ph.D. degree in mechanical and aerospace engineering from Princeton University, Princeton, NJ, in 2012. He is an Assistant Professor of mechanical engineering and a data science fellow with the eScience institute at the University of Washington.



**J. Nathan Kutz** received the B.S. degrees in physics and mathematics from the University of Washington, Seattle, WA, in 1990, and the Ph.D. degree in applied mathematics from Northwestern University, Evanston, IL, in 1994. He is currently a Professor of applied mathematics, adjunct professor of physics and electrical engineering, and a senior data science fellow with the eScience institute at the University of Washington.