Allocation Strategies for High Fidelity Models in the Multifidelity Regime[*]

Daniel J. Perry[†], Robert M. Kirby[†], Akil Narayan[‡], and Ross T. Whitaker[†]

Abstract. We propose a novel approach to allocating resources for expensive simulations of high fidelity models when used in a multifidelity framework. Allocation decisions that distribute computational resources across several simulation models become extremely important in situations where only a small number of expensive high fidelity simulations can be run. We identify this allocation decision as a problem in optimal subset selection and subsequently regularize this problem so that solutions can be computed. Our regularized formulation yields a type of group lasso problem that has been studied in the literature to accomplish subset selection. Our numerical results compare the performance of algorithms that solve the group lasso problem for algorithmic allocation against a variety of other strategies, including those based on classical linear algebraic pivoting routines and those derived from more modern machine learning–based methods. We demonstrate on well-known synthetic problems and more difficult real-world simulations that this group lasso solution to the relaxed optimal subset selection problem performs better than the alternatives.

Key words. multifidelity models, column subset selection, mixed norm regularization

AMS subject classification. 65D05

DOI. 10.1137/17M1144714

1. Introduction. A persistent challenge in uncertainty quantification is estimating the effect of uncertain parameters on quantities of interest. A common approach to understanding the effect of a parameter is to evaluate an ensemble of simulations for various parameter values. This approach is reasonable when multiple runs of a simulation model or experiment are easily obtained. Unfortunately, a simulation model or discretization that is more true-to-life, or has *higher fidelity*, requires additional computational resources due to, for example, an increased number of discrete elements or more expensive modeling of complex phenomena. For these reasons, a *high-fidelity* model simulation can incur significant computation cost, and repeating such a simulation a sufficient number of times to understand parameter effects can quickly become infeasible.

[†]The Scientific Computing and Imaging Institute (SCI) Institute, University of Utah, Salt Lake City, UT 84112 (dperry@cs.utah.edu, kirby@cs.utah.edu, whitaker@sci.utah.edu).
[‡]The Scientific Computing and Imaging Institute (SCI), and Department of Mathematics, University of Utah, Salt Lake City, UT 84112 (akil@sci.utah.edu).

Recent work has introduced an effective solution to this problem by using multiple fidelities of simulation models where less-expensive, lower-fidelity versions of the high-fidelity model are used to learn the parametric structure of a simulation. This structure is utilized to choose a small parameter ensemble at which high-fidelity runs are assembled, providing insight into the finer details and effects of the simulation parameters [23, 38]. One important decision in this allocation process is which parameter values should be used in the less costly low- and medium-fidelity simulations and which values are worth the more costly high-fidelity simulation. This decision becomes extremely important for situations where it is physically impossible to run the high-fidelity simulation more than a small number of times, say $\mathcal{O}(10)$ times.

By assuming the simulation realizations are elements in a Hilbert space, previous work uses a Gram matrix associated to a parameter ensemble to learn the parametric structure. Standard linear algebra tools, such as the Cholesky decomposition, are utilized to rank "important" parameter values where high-fidelity simulations are run.

The multifidelity situation is special among subset selection problems since we cannot add additional elements to a high-fidelity subset due to the significant marginal cost of additional computation. Thus choosing the *best* subset becomes quite critical in order to effectively utilize a very small number of high-fidelity simulations. While the classical linear algebraic approaches mentioned previously perform reasonably well, these approaches ignore the extensive work done in subset selection in a more general context in both the data analysis and the machine learning literature [10, 5, 3, 4, 25, 1].

Our main contributions in addressing this problem are as follows:
- Our work includes substantial experimental comparisons with a broad range of subset selection algorithms. These include classical methods, such as randomized sampling, single-layer Gaussian processes, and pivoted numerical linear algebra routines, along with more modern machine learning–based strategies, such as leverage sampling and neural networks. We conclude that a particular group orthogonal matching pursuit (GOMP) algorithm yields superior results for the problems we consider.
- We apply existing GOMP approximation theory to a single-fidelity approximation problem, providing justification for the competitive performance of GOMP compared to other algorithms in the context of subset selection. Our analysis does not directly apply to the multifidelity method we investigate but provides a first step toward understanding group matching pursuit approaches for this multifidelity strategy.
- We empirically show that the GOMP multifidelity procedure is effective when applied to a variety of nontrivial large-scale problems in computational science, such as compressible fluid dynamics and structural topology optimization. In particular, we observe that GOMP-based methods yield superior results in almost every situation we have tried.

We note that there are many alternative approaches to tackling the multifidelity problem, and these alternatives are substantially different in scope, applicability, and goals. For example, the authors of [24] use additive corrections to combine high-fidelity and low-fidelity quantities of interest into predictions; the authors of [19, 28] use a multilevel Gaussian process strategy to build Gaussian process predictors from models at different hierarchies; and the authors of [26] use a multilevel Monte Carlo prediction strategy for expectations of scalar quantities of

interest to devise a resource allocation strategy. In contrast to these approaches, our strategy uses low-fidelity parametric variation to guide allocation of high-fidelity effort but does not require any transformation operators between levels (such as, e.g., additive corrections). A direct comparison between our strategy and alternative multifidelity approaches is beyond the scope of this paper, and so we leave such a comparison for future work.

2. Previous work. We will now consider the multifidelity proxy model introduced in [23] and further studied in [38]. The authors of [23] proposed using samples from a low-fidelity model to inform the parameter selection of the much more costly high-fidelity samples and to compute the reconstruction weights of the high-fidelity proxy model based on those low-fidelity samples. The weights are computed from the Gramian (also known as the Gram matrix or inner product matrix) of the low-fidelity samples, and so the resulting model can be considered a nonparametric model with weights derived from the low-fidelity Gramian. For simplicity, we will consider only a two-level ("bifidelity") situation with only a single low- and high-fidelity model. However, this procedure can be applied to many levels [23].

2.1. Gramian nonparametric multifidelity algorithm. We consider the following general multifidelity setup: Let $z \in D \subset \mathbb{R}^q$, $q \geq 1$, be a common parametric input into two simulations models of differing fidelities. We use u^L and u^H to denote the outputs of the low- and high-fidelity models, respectively, with $u^L(z)$ a vector-valued output of the low-fidelity model evaluated at parameter value z. The models produce outputs in vector spaces V^L and V^H, respectively. To summarize our notation,

$$u^L : D \to V^L, \qquad u^H : D \to V^H.$$

The vectors u^L and u^H are assumed to represent any spatial/temporal effects of interest. The multifidelity regime occurs when u^H is more faithful to reality than u^L but requires greater computational effort to simulate. That is,

$$\text{(1)} \qquad \left\| u^H(z) - u(z) \right\| \ll \left\| u^L(z) - u(z) \right\|, \qquad \text{Cost}\left(u^H(z) \right) \gg \text{Cost}\left(u^L(z) \right),$$

where $u(z)$ represents reality,[1] and $\text{Cost}(\cdot)$ is the requisite computational burden of evaluating the argument. In practice, the assumptions (1) frequently imply that the dimensions of the vector spaces satisfy

$$\dim V^L = d^L \ll d^H = \dim V^H,$$

but it is not necessary to assume this and the multifidelity algorithm does not require this assumption. A concrete example of the above situation is the following: Consider Poisson's equation on a domain $\Omega \subset \mathbb{R}^2$:

$$-\triangle_x u(x, z) = f(x, z), \qquad x \in \Omega, \;\; z \in D,$$

---

[1] We are intentionally being vague in defining $u(z)$ and the norms in (1). Such relations can be established through formal means, such as convergence of discretizations of mathematical models, or through more informal means, such as expert belief or knowledge in superiority of the high-fidelity model.

where $\triangle_x$ denotes the Laplacian with respect to the x variable. One may compute appropriate solutions to this problem with finite element methods and suppose that u^L and u^H are two finite element solutions computed on different meshes: u^L is computed on a relatively coarse mesh with mesh size parameter T, and u^H is computed on a relatively fine mesh with mesh size parameter $t \ll T$. Since the total number of elements on a mesh in two dimensions scales like $1/T^2$ or $1/t^2$, this implies that the number of degrees of freedom for u^H, $d^H \sim 1/t^2$, is much larger than that for u^L, $d^L \sim 1/T^2$. Assuming the polynomial degree p of approximation within each element is the same on both meshes, a standard a priori finite element analysis yields the estimate

$$\left\| u - u^L \right\|_{L^2(\Omega)} \leq CT^{p+1}, \qquad \left\| u - u^H \right\|_{L^2(\Omega)} \leq Ct^{p+1}$$

for a constant C. Since $t \ll T$, and assuming the estimate above is sharp, this is an instance of the situation (1), where the unspecified norm in that equation is the L^2 norm on Ω, and the cost function refers to computational cost. However, we note that our procedure applies to the more abstract situation (1) and our procedure is not limited to a particular discretization, e.g., finite element methods.

Due to the high computational cost, the number of high-fidelity simulations is limited to m, with $m = \mathcal{O}(10)$ being a common bound. In contrast, the low-fidelity model is much more computationally affordable, with $n \gg m$ low-fidelity simulations available.

Let $\gamma = \{z_1, \ldots, z_n\}$ be a set of sample points in D, which define matrices

$$\boldsymbol{a}_j^L = u^L(z_j), \qquad \boldsymbol{A}^L = \left[\begin{array}{cccc} \boldsymbol{a}_1^L & \boldsymbol{a}_2^L & \ldots & \boldsymbol{a}_n^L \end{array}\right] \in \mathbb{R}^{d^L \times n},$$
$$\boldsymbol{a}_j^H = u^H(z_j), \qquad \boldsymbol{A}^H = \left[\begin{array}{cccc} \boldsymbol{a}_1^H & \boldsymbol{a}_2^H & \ldots & \boldsymbol{a}_n^H \end{array}\right] \in \mathbb{R}^{d^H \times n}.$$

Let $S \subset [n]$ denote a generic set of column indices, and for $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ having columns $\boldsymbol{a}_j$ we define

$$S = \left\{j_1, \ldots, j_{|S|}\right\}, \qquad \boldsymbol{A}_S = \left[\begin{array}{cccc} \boldsymbol{a}_{j_1} & \boldsymbol{a}_{j_2} & \ldots & \boldsymbol{a}_{j_{|S|}} \end{array}\right].$$

The authors of [23] showed that the structure of $\boldsymbol{A}^L$ can be used to identify a small number of column indices $S \subset [n]$, $|S| = m$, so that $\boldsymbol{A}_S^L$ can be used to form a rank-m approximation to $\boldsymbol{A}^L$, and $\boldsymbol{A}_S^H$ can be used to form a rank-m approximation to $\boldsymbol{A}^H$. Precisely, they form the approximations

$$(2) \qquad\qquad \boldsymbol{A}^L \approx \boldsymbol{A}_S^L \left(\boldsymbol{A}_S^L\right)^\dagger \boldsymbol{A}^L, \qquad \boldsymbol{A}^H \approx \boldsymbol{A}_S^H \left(\boldsymbol{A}_S^L\right)^\dagger \boldsymbol{A}^L,$$

where $\boldsymbol{A}^\dagger$ is the Moore–Penrose pseudoinverse of $\boldsymbol{A}$. An important observation in the above approximations is that the representation for $\boldsymbol{A}^H$ requires *only* $\boldsymbol{A}_S^H$, i.e., it only requires $m = |S|$ evaluations of the high-fidelity model. The construction of S is performed in a greedy fashion, precisely as the first m ordered pivots in a pivoted Cholesky decomposition of $\left(\boldsymbol{A}^L\right)^T \boldsymbol{A}^L$. While $\boldsymbol{A}^H$ only represents u^H on a discrete set γ, the procedure above forms the approximation

$$(3) \qquad\qquad u^H(z) \approx \sum_{i=1}^{m} c_i(z) u^H(z_{j_i}),$$

where the coefficients c_i are the expansion coefficients of $u^L(z)$ in a least-squares approximation with the basis $\{u^L(z_{j_i})\}_{i=1}^m$. Thus, the above approximation to the high-fidelity simulation may actually be evaluated at any location $z \in D$ if $u^L(z)$ is known. This procedure has the following advantages:

- Once m high-fidelity simulations have been computed and stored, the mathematical construction of the approximation (3) to the high-fidelity model is tantamount to evaluating the coefficients $c_i(z)$, the latter of which has computational complexity depending only on the low-fidelity model; cf. (3). Explicitly constructing the vector $u^H(z) \in \mathbb{R}^{d^H}$ naturally requires an m-fold addition of length-d^H vectors as in (3).
- The subset S is identified via analysis of the inexpensive low-fidelity model, so that a very large set γ may be used to properly capture the parametric variation over D.
- It is not necessary for the spatiotemporal features of the low-fidelity to mimic those of the high-fidelity model. Section 6.2 in [23] reveals that u^L and u^H may actually be entirely disparate models, yet the approximation (3) can be accurate. However, these are empirical observations, and a deeper understanding of precisely how disparate low- and high-fidelity models can be is an open problem.
- The cost of evaluating (3) can be split into two portions: an "offline" stage, where the ensemble $\{u^N(z_{j_i})\}_{i=1}^m$ is compiled and stored, and an "online" stage, where the coefficients c_i must be computed for a given z. The offline stage requires m high-fidelity simulations and $n \gg m$ low-fidelity model evaluations, and is thus expensive; the online stage requires only a single low-fidelity model evaluation, $u^L(z)$, from which the coefficients c_i are computed. Thus, this procedure can be efficient when one requires an approximation to the high-fidelity model at $M \gg m$ values of z.

2.2. Subset selection. The critical portion of the previous section's algorithm is the identification of the column subset S. In order to guarantee convergence of the high-fidelity approximation in an appropriate norm on V^H, one must also have some correlation between the *parametric* variation of the high- and low-fidelity models. Theorem 4.4 of [23] gives the conditions which can guarantee quantitative proximity of the multifidelity surrogate and the high-fidelity model. These conditions are, admittedly, difficult to verify. In this paper, we assume such parametric correlation exists and focus exclusively on the problem of identification of S. Abstractly, this is a problem of subset selection among the n columns of $\boldsymbol{A}^L$, and thus this problem focuses entirely on the low-fidelity model. To emphasize this and to simplify notation, we dispense with the L and H superscripts, hereafter writing

$$\boldsymbol{A} \leftarrow \boldsymbol{A}^L, \qquad \boldsymbol{a}_j \leftarrow \boldsymbol{a}_j^L.$$

The subset selection problem, identification of S, is an allocation problem. That is, given an accurate but expensive high-fidelity simulation, how do we allocate high-fidelity computational resources across the set of possible simulations γ? Noting the approximations (2), one way to accomplish this is to choose S in a way that captures as much variation in the low-fidelity model as possible.

Definition 2.1. *The column subset selection problem (CSSP). Find the column subset matrix* $\mathbf{C} = \mathbf{AS}$, *where* $\mathbf{C} \in \mathbb{R}^{d \times m}$, $\mathbf{S} \in \mathbb{R}^{n \times m}$ *is a column selection matrix, and the* $m \leq n$

columns of $\mathbf{C}$ *are a subset of the n columns of* $\mathbf{A}$*, so that*

$$(4) \qquad\qquad r = \|\mathbf{R}\|_F^2 = \|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\|_F^2$$

is minimized.

The matrix $\boldsymbol{S}$ above mirrors the subset S discussed earlier, and $\boldsymbol{C} = \boldsymbol{A}_S$. The exact CSSP is at present conjectured to be NP-complete [31] and, if this were true, requires the evaluation of all subsets of size m. However, there have been a large variety of strategies proposed to address the CSSP; we summarize a selection of these strategies below:

- *Leverage sampling* [4, 25, 11, 29]. The singular value decomposition (SVD) of $\mathbf{A}$ is computed or approximated, and this information is used to choose a column subset of $\boldsymbol{A}$, either randomly or deterministically. Leverage sampling has attractive theoretical guarantees, e.g., $\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$, where $\mathbf{C}$ is chosen according to the leverage score and m is $\mathcal{O}(1/\epsilon)$. (The best rank-k matrix $\boldsymbol{A}_k$ is defined by the Eckart–Young–Mirsky theorem.) In our results section, we observe that leverage sampling on our datasets is less competitive than other sampling techniques.
- *The CUR matrix decomposition* [21]. This performs a column ($\mathbf{C}$) and row ($\mathbf{R}$) subset selection on $\boldsymbol{A}$, resulting in an element-based decomposition of a matrix, so that $\mathbf{A} \approx \mathbf{C}\mathbf{U}\mathbf{R}$, where $\mathbf{U} = \mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger$. The best known CUR matrix error bounds are obtained by using leverage sampling for the column and row subset selection [21]; for that reason, we consider our experimental comparison to leverage sampling and random sampling as sufficient to characterize related methods such as the CUR decomposition technique.
- *Deterministic interpolative decompositions (IDs)* [10]. These are similar in spirit to CUR; ID methods frequently use a Gram–Schmidt-based column-pivoted QR method for subset selection, resulting in an approximation $\mathbf{A} \approx \mathbf{C}\mathbf{B}$, where $\mathbf{C}$ is a column subset of $\mathbf{A}$ and $\mathbf{B}$ is a coefficient matrix that minimizes the approximation error. (For example, $\boldsymbol{B} = \boldsymbol{C}^\dagger\boldsymbol{A}$ is a possible choice for $\boldsymbol{B}$.) In this paper, we will use the pivoted QR method as an exemplar of deterministic ID algorithms.
- *Pivoted QR decompositions* [13, 15]. One historical use of these routines was to select a column subset in order to reliably compute the least-squares solution of a linear system. For the CSSP, we run a pivoted QR routine and then sample the columns in order identified by the pivots [9, 25]. QR-based subset selection has a much worse known theoretical guarantee with respect to the rank-k tail bound $\|\mathbf{A} - \mathbf{A}_k\|_F^2$ than leverage sampling, but empirically we have observed that it performs much better. The authors of [23, 38] propose using a pivoted Cholesky routine in a manner that is algebraically equivalent to a pivoted QR method. Alternative linear algebraic pivoting strategies can be used as approaches to the CSSP, such as partial or full pivoted LU [14]. However, besides their mention in [38] as a possible direction, we have not observed their use in existing CSSP work.

We note that [1] showed a different type of bound for a general greedy strategy, of which the pivoted QR, LU, and Cholesky methods are specific instances. The theoretical bound shown for the general greedy strategy is $\|\mathbf{C}\mathbf{C}^\dagger\mathbf{A}\|_F^2 \geq (1-\epsilon)\|\mathbf{D}\mathbf{D}^\dagger\mathbf{A}\|_F^2$, where $\mathbf{D}$ is the optimal subset with rank($\mathbf{D}$) = k (defined by Definition 2.1), and $\mathbf{C}$ is a computed rank-m approximation,

where $m \sim 16k$. Table 1 below gives more precise estimates. This bound is also of interest because large values of $\|\mathbf{CC}^\dagger\mathbf{A}\|_F^2$ are related to small values of $\|\mathbf{A} - \mathbf{CC}^\dagger\mathbf{A}\|_F^2$. Working in the matrix Frobenius norm $\|\cdot\|_F$ is a common choice since this generally leads to easier analysis. The analysis in this manuscript can be used to provide estimates in the Frobenius norm; we expect estimates in other norms to be significantly more complicated and leave investigation of other norms to future work.

All the strategies above are described as applied to a set of Euclidean vectors, but each only requires a set of elements in a Hilbert space. Consequently, by using the same Hilbert space assumptions from prior work [23], we are able to make use of a broad class of CSSP algorithms for the allocation portion of the multifidelity procedure under consideration.

3. Allocation strategies for multifidelity simulation. In this section, we start by reformulating a regularized version of the CSSP as a mixed-norm constrained least-squares problem. This relaxation of the problem opens up to us a variety of different solution techniques. We then present some theoretical justifications for why GOMP is competitive for the CSSP. We follow this by a theoretical discussion of how various methods we investigated compared, against each other and to GOMP, in terms of error guarantees, number of columns, and asymptotic runtime to achieve those guarantees.

3.1. Relaxation of the exact subset problem. We now reformulate a regularized version of the CSSP as a mixed-norm constrained least-squares problem. The goal of the discussion below is to write a relaxed version of the residual minimization for the CSSP as an optimization problem over a residual; such a formulation naturally suggests the usage of group matching algorithms for residual minimization, where groups are matrix columns.

Each column of the subset matrix $\mathbf{C}$ is taken from the full dataset, and therefore $\mathbf{c}_j \in \{\mathbf{a}_i | i = 1, \dots n\}$. This representation can be formulated so that the residual matrix can be expressed as $\mathbf{R} = \boldsymbol{A} - \mathbf{C}\tilde{\mathbf{B}}$, where $\tilde{\mathbf{B}} \in \mathbb{R}^{m \times n}$ is a coefficient matrix that essentially combines the coreset elements, or atoms, via weights. Each data point is therefore approximated as

$$(5) \qquad \qquad \mathbf{a}_i \approx \mathbf{C}\tilde{\mathbf{b}}_i,$$

and the residual is

$$(6) \qquad r = \|\boldsymbol{R}\|_F^2 = \sum_i \|\mathbf{a}_i - \mathbf{C}\tilde{\mathbf{b}}_i\|_2^2 = \|\mathbf{A} - \mathbf{C}\tilde{\mathbf{B}}\|_F^2 = \|\mathbf{A} - \mathbf{AB}\|_F^2,$$

where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is given by $\boldsymbol{B} = \boldsymbol{S}\tilde{\boldsymbol{B}}$. In other words, $\boldsymbol{B}$ is formed from $\tilde{\boldsymbol{B}}$ by inserting rows of zeros associated to columns of $\boldsymbol{A}$ that are not chosen in $\boldsymbol{C}$. The smaller matrix $\tilde{\mathbf{B}}$ is a submatrix of the larger, more sparse $\mathbf{B}$, where m specific (nonzero) rows are preserved.

Instead of building $\mathbf{B}$ from $\tilde{\mathbf{B}}$, we are interested in solving for $\mathbf{B}$ directly, which minimizes r. The optimum of the objective $\|\mathbf{A} - \mathbf{AB}\|_F^2$ would result in $\mathbf{B} = \mathbf{I}$; however, if we constrain or regularize $\mathbf{B}$ so that it is row-sparse, as described above, then we can compute a matrix $\mathbf{B}$ with the desired properties. This leads to the regularized formulation for column subset selection,

$$(7) \qquad \arg\min_{\mathbf{B}\in\mathbb{R}^{n \times n}} \|\mathbf{A} - \mathbf{AB}\|_F^2 + \lambda\|\mathbf{B}^T\|_{0,0},$$

where $\|\mathbf{M}\|_{0,0} = \sum_i \|\mathbf{M}_{:,i}\|_0$ is a mixed $\ell_{0,0}$ norm that induces column sparsity. Note the equivalence of the objective in (7) to the CSSP objective in (4) above, where $\mathbf{AB} = \mathbf{CC}^{\dagger}\mathbf{A}$, and the columns selected for $\mathbf{C}$ correspond to the nonzero rows of $\mathbf{B}$. Because of the mixed $\ell_{0,0}$ norm, we still have an NP-complete problem. However, in this new form, we can relax the mixed norm penalty to an $\ell_{2,1}$ norm, where $\|\mathbf{M}\|_{2,1} = \sum_i \|\mathbf{M}_{:,i}\|_2$:

$$(8) \qquad \mathbf{B}^* = \arg \min_{\mathbf{B} \in \mathbb{R}^{n \times n}} \|\mathbf{A} - \mathbf{AB}\|_F^2 + \lambda \|\mathbf{B}^T\|_{2,1}.$$

This relaxed form also induces row sparsity of $\boldsymbol{B}$ and is more amenable to numerical methods for its solution; see, e.g., [36, 35]. We have made two relaxations of the CSSP: the first is the introduction of λ, and the second relaxes the mixed $\ell_{0,0}$ norm to an $\ell_{2,1}$ norm. In general, we therefore neither expect the solution to (8) to match the solution of (7), nor do we expect that it recovers the optimal CSSP solution. However, in Theorems 3.1 and 3.2 below, we give sufficient conditions for specialized cases where this relaxation recovers the CSSP solution.

The type of problem shown in (8) is known in the optimization literature as a *group lasso* problem [36]. Group lasso problems have been well studied, and there are a variety of algorithms available for their solution. For the results in this paper, we use GOMP. Although this is a greedy algorithm and not optimal in general, it performs well for selecting subsets of columns [20]. Two other methods that have demonstrated efficiency in solving this type of optimization are the *group least angle regression* (GLARS) method and the *alternating direction method of multipliers* (ADMM) [36, 6, 12]. We have found GOMP and GLARS to be the most useful overall for the exploration of various coreset sizes, while ADMM is well suited to extremely large datasets [12, 7]. The GOMP algorithm is simpler than GLARS, and since numerical tests on our particular datasets suggest no predictive advantage of one over the other, we have opted to concentrate on GOMP here. Some discussion that compares these differences between some of these group optimization problems is present in [36], but we are unaware of quantitative results comparing how well approximate solvers for (8) perform.

While GOMP can be computed over a discrete data matrix $\boldsymbol{A}$, it only really makes use of the inner products of the data vectors, so that the algorithm only assumes a Hilbert space over the data points. This means that the approach can be used in the context of continuous simulation vectors $u^L(z_i)$ as long as they exist in a proper Hilbert space. A pseudocode presentation of GOMP for column subset selection with continuous vectors is shown in Algorithm 1. The algorithm takes the input vectors, $u(\gamma)$, as well as two parameters, λ and ϵ, to control the sparsity and the precision, respectively (Algorithm 1, line 1). We initialize the coefficient matrix, $\mathbf{B}$, to be the zero matrix and the active set, $\mathcal{A}$, to be empty (line 4). Now, until the sparsity level or the error precision is satisfied, we loop and do the following in each iteration: (re)compute the residual matrix based on the current coefficient matrix (line 7), recompute the group correlations based on the residual matrix (line 8), find the unused element with maximum group correlation and add it to the active set (lines 9 and 13), and recompute the coefficient matrix (line 14). Note that the inner product matrix $\mathbf{Q}$ is computed using the appropriate inner product for the space. The residual matrix $\mathbf{R}$ is actually the residual-correlation matrix, as it represents the inner product of the residual with the input functions. For example, considering line 7, each entry $\mathbf{R}_{i,j} = \langle u(z_i), r(z_j) \rangle = \langle u(z_i), u(z_j) - \sum_{k=1}^N b_{j,k} u(z_k) \rangle$, where $r(z_j)$ represents the residual at sample point z_j. The

---

Algorithm 1 Group orthogonal matching pursuit for subset selection of continuous vectors.

1: **Input** $u(\gamma)$ - input vectors, λ - sparsity parameter, ϵ - precision parameter
2: **Output** $\mathbf{B} \in \mathbb{R}^{m \times p}$, coefficient matrix, $\mathcal{A}$, the subset indices
3: **procedure** $\text{GOMP}(u(\gamma), \lambda, \epsilon)$
4:      Set $\mathbf{Q}_{ij} = \langle u(z_i), u(z_j) \rangle$, the inner product matrix
5:      Initialize $\mathcal{A} = \emptyset$, $\mathbf{B} = \mathbf{0}$
6:      **while** $\|\mathbf{B}^T\|_{2,1} < \frac{1}{\lambda}$ **do**
7:          $\mathbf{R} = \mathbf{Q} - \mathbf{QB}$ $\triangleright$ update residual
8:          $\mathbf{c}_i = \|\mathbf{R}_i\|_2$
9:          $i^* = \arg\max_i \mathbf{c}_{\mathcal{A}^c}$ $\triangleright$ element with max group correlation
10:         **if** $\mathbf{c}_{i^*} \leq \epsilon$ **then**
11:             Break
12:         **end if**
13:         $\mathcal{A} = \mathcal{A} \cup \{i^*\}$
14:         $\mathbf{B}_{\mathcal{A}} = \mathbf{Q}_{\mathcal{A},\mathcal{A}}^{\dagger} \mathbf{Q}_{\mathcal{A},:}$ $\triangleright$ recompute coefficient matrix
15:      **end while**
16:      **return** $\mathbf{B}, \mathcal{A}$
17: **end procedure**

---

sparsity parameter λ only controls the stopping criterion and is not used in the subset selection procedure.

3.2. Group orthogonal matching pursuit. We now present some theoretical justification for why GOMP is competitive for the CSSP. Our two results are Theorems 3.1 and 3.2. In particular, Theorem 3.2 shows that GOMP can identify a small representative column subset under some assumptions on the matrix and the representative set.

Let $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ be a matrix whose n columns consist of snapshot or feature vectors. We assume that the columns of $\boldsymbol{A}$ have unit norm, i.e., if $\boldsymbol{a}_j$ is the jth column of $\boldsymbol{A}$, then we assume that

$$(9) \qquad \|\boldsymbol{a}_j\|_2 = 1.$$

Our goal is to construct a low-rank approximation to every feature (column) in $\boldsymbol{A}$ using linear combinations of a (small) subset of feature vectors in $\boldsymbol{A}$. As described in the previous section, our strategy for accomplishing this is to solve the following optimization problem for the matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$:

$$(10) \qquad \min \|\boldsymbol{A} - \boldsymbol{AB}\|_F^2 + \lambda \|\boldsymbol{B}^T\|_{2,1},$$

where

$$\|\boldsymbol{B}^T\|_{2,1} = \sum_{j=1}^{n} \|\boldsymbol{b}_{j,:}\|_2,$$

and $\boldsymbol{b}_{j,:}$ is the jth row of $\boldsymbol{B}$. The mixed norm $\|\cdot\|_{2,1}$ promotes column sparsity of its argument. In the context of the product $\boldsymbol{AB}$, column sparsity of $\boldsymbol{B}^T$ implies that a small number of columns of $\boldsymbol{A}$ are used to approximate the remaining columns.

Because we solve (10) using a GOMP algorithm, we proceed to rewrite this problem in the language of GOMP algorithms. Let $\boldsymbol{a}_i$, $i \in [n]$, denote the ith column of $\boldsymbol{A}$. We call a set of (column) indices $S \subset [n]$ a *basis* set for $\boldsymbol{A}$ if

$$\mathrm{span}\left\{\boldsymbol{a}_i \mid i \in S\right\} = \mathrm{range}\left(\boldsymbol{A}\right).$$

Basis sets are not unique in general. A *complement* set of indices equals $[n]\backslash S$ for a given basis set S. Our first result is a consistency result: given a basis set that satisfies reasonable assumptions, GOMP can identify this set.

Theorem 3.1. *Let $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ have columns $\boldsymbol{a}_i \in \mathbb{R}^d$, and let $S_{\mathrm{g}} \subset [n]$ be a basis set for $\boldsymbol{A}$. For each $j \in [n]$, there is a unique expansion of $\boldsymbol{a}_j$ in columns of the basis set:*

$$\text{(11a)} \qquad \boldsymbol{a}_j = \sum_{i \in S_{\mathrm{g}}} D_{i,j} \boldsymbol{a}_i.$$

Use the elements $D_{i,j}$ to define the matrix $\boldsymbol{D}$. If both conditions

$$\text{(11b)} \qquad \bar{D} := \max_{j \in [n]\backslash S_{\mathrm{g}}} \sum_{i \in S_{\mathrm{g}}} |D_{i,j}| < 1, \qquad \epsilon > 0,$$

are satisfied, then GOMP, i.e., Algorithm 1, terminates after at most $|S_{\mathrm{g}}|$ steps. At termination, the algorithm identifies a subset of S_{g} if fewer than $|S_{\mathrm{g}}|$ steps are taken and identifies S_{g} if $|S_{\mathrm{g}}|$ steps are taken.

Proof. For any matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of appropriate size, the following identity holds:

$$\mathrm{vec}\left(\boldsymbol{B}^T \boldsymbol{A}^T\right) = \left(\boldsymbol{A} \otimes \boldsymbol{I}_n\right) \mathrm{vec}\left(\boldsymbol{B}^T\right),$$

where $\otimes$ is the Kronecker product, and $\mathrm{vec}\left(\cdot\right)$ is the vectorization (vertical concatenation of columns) of its argument. Defining $\boldsymbol{g} := \mathrm{vec}\left(\boldsymbol{B}^T\right)$, $\boldsymbol{f} := \mathrm{vec}\left(\boldsymbol{A}^T\right)$, and $\boldsymbol{H} = \left(\boldsymbol{A} \otimes \boldsymbol{I}_n\right)$, we can write our desired relation $\boldsymbol{A} \approx \boldsymbol{AB}$ in vectorized format:

$$\text{(12)} \qquad \boldsymbol{Hg} \approx \boldsymbol{f}.$$

The goal of (10), aiming to promote row sparsity of $\boldsymbol{B}$, is now translated into a group sparsity of entries of $\boldsymbol{g} = \mathrm{vec}\left(\boldsymbol{B}^T\right)$. Each group of elements in $\boldsymbol{g}$ corresponds to an individual row of $\boldsymbol{B}$. We seek to prove that GOMP applied to this system recovers the basis groups. Our main tool to accomplish this is Corollary 1 in [20]; this corollary shows that a version of GOMP in that reference identifies the correct groups. We will first show this for our case and then explain how this can be used to conclude the identification of correct groups for Algorithm 1.

We will let G denote a generic subset of $[n^2]$. We define the groups $G_i \subset [n^2]$, $i = 1, \ldots, n$, as sets of n sequential indices,

$$G_i = \left\{n(i-1)+1, n(i-1)+2, \ldots, ni\right\},$$

and thus $G_1, \ldots, G_n$ partition $[n^2]$. We denote by $\boldsymbol{H}_G$ the matrix $\boldsymbol{H}$ restricted to the G-indexed columns. (Recall that G is a subset of $[n^2]$ and H has n^2 columns.) The basis columns S_{g} can be translated into a set of basis groups G_{g}:

$$S_{\mathrm{g}} \subset [n] \implies G_{\mathrm{g}} \subset [n^2], \qquad G_{\mathrm{g}} = \cup_{i \in S_{\mathrm{g}}} G_i.$$

The complement of S_{g} in $[n]$, and the associated set G_{d}, are defined accordingly:

$$S_{\mathrm{d}} := [n] \backslash S_{\mathrm{g}}, \qquad G_{\mathrm{d}} = \cup_{i \in S_{\mathrm{d}}} G_i.$$

We use $\boldsymbol{A}_S$ to denote selection of the S-indexed columns of $\boldsymbol{A}$ and $\boldsymbol{H}_G$ to denote selection of the G-indexed columns of $\boldsymbol{H}$. Under the assumption (9), we have $\boldsymbol{H}_{G_j}^T \boldsymbol{H}_{G_j} = \boldsymbol{I}_n$ for each group $j = 1, \ldots, n$.

Row sparsity of $\boldsymbol{B}$ means that we wish to choose a small (sparse) number of groups G_j. Our next step is to show that the GOMP algorithm in [20] applied to (12) identifies groups of indices associated to the column indices S_{g}. Consider the matrix

$$\boldsymbol{M} = \boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}.$$

Since S_{g} is a set of basis indices, then $\boldsymbol{M}$ is invertible. This implies

$$\boldsymbol{H}_{G_{\mathrm{g}}}^{\dagger} = \left(\boldsymbol{A}_{S_{\mathrm{g}}} \otimes \boldsymbol{I}_n \right)^{\dagger} = \boldsymbol{A}_{S_{\mathrm{g}}}^{\dagger} \otimes \boldsymbol{I}_n = \left(\boldsymbol{M}^{-1} \boldsymbol{A}_{S_{\mathrm{g}}}^T \right) \otimes \boldsymbol{I}_n.$$

Then

$$\boldsymbol{H}_{G_{\mathrm{g}}}^{\dagger} \boldsymbol{H}_{G_{\mathrm{d}}} = \left(\left(\boldsymbol{M}^{-1} \boldsymbol{A}_{S_{\mathrm{g}}}^T \right) \otimes \boldsymbol{I}_n \right) \left(\boldsymbol{A}_{S_{\mathrm{d}}} \otimes \boldsymbol{I}_n \right) = \left(\boldsymbol{M}^{-1} \boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{d}}} \right) \otimes \boldsymbol{I}_n.$$

Define $\boldsymbol{D} := \boldsymbol{M}^{-1} \boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{d}}} \in \mathbb{R}^{s \times (n-s)}$, and note that it is comprised of the entries $D_{i,j}$ defined in (11a). We need to introduce a mixed vector norm. Let $q \in \mathbb{N}$ be arbitrary; for a vector $\boldsymbol{x} \in \mathbb{R}^{qn}$, we define

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_q \end{pmatrix} \in \mathbb{R}^{qn}, \quad \boldsymbol{x}_j \in \mathbb{R}^n \quad \implies \quad \|\boldsymbol{x}\|_{2,1} = \sum_{j=1}^{q} \|\boldsymbol{x}_j\|_2 \, .$$

Given such a vector $\boldsymbol{x}$, we use notation $\boldsymbol{x}_j$ to mean a length-n vector from the elements of $\boldsymbol{x}$ as above. We have

$$\|(\boldsymbol{D} \otimes \boldsymbol{I}_n) \, \boldsymbol{v}\|_{2,1} = \left\| \sum_{j=1}^{n-s} (\boldsymbol{d}_j \otimes \boldsymbol{v}_j) \right\|_{2,1} \leq \sum_{j=1}^{n-s} \|\boldsymbol{d}_j \otimes \boldsymbol{v}_j\|_{2,1}$$

$$= \sum_{j=1}^{n-s} \sum_{i=1}^{s} |D_{i,j}| \, \|\boldsymbol{v}_j\|_2 \leq \left(\max_{1 \leq j \leq n-s} \sum_{i=1}^{s} |D_{i,j}| \right) \sum_{j=1}^{n-s} \|\boldsymbol{v}_j\|_2 = \|\boldsymbol{D}\|_1 \, \|\boldsymbol{v}\|_{2,1}.$$

Chaining together these results with the assumption (11b), we have shown that

$$\left\| \boldsymbol{H}_{G_{\mathrm{g}}}^{+} \boldsymbol{H}_{G_{\mathrm{d}}} \right\|_{*2,1} := \sup_{\boldsymbol{v} \in \mathbb{R}^{(n-s)n} \backslash \{\boldsymbol{0}\}} \frac{\left\| (\boldsymbol{D} \otimes \boldsymbol{I}_n)\, \boldsymbol{v} \right\|_{2,1}}{\left\| \boldsymbol{v} \right\|_{2,1}} < 1,$$

where we have used $\|\cdot\|_{*2,1}$ to denote the norm induced by the $\|\cdot\|_{2,1}$ norm on vectors, and not the mixed $\ell_{2,1}$ elementwise norm on matrices. Under this condition, Corollary 1 in [20] shows that the GOMP algorithm in that reference applied to $\boldsymbol{Hg} \approx \boldsymbol{f}$ identifies the basis groups in $\boldsymbol{d}$, equivalently that the GOMP algorithm identifies the basis columns of $\boldsymbol{A}$. This is *almost* the result we require: Our GOMP method, Algorithm 1, is almost identical to the algorithm in [20], the only difference being the presence of an additional regularization parameter λ that influences the stopping criterion. Without this criterion, Algorithm 1 coincides with the algorithm in [20]. Therefore, Algorithm 1 selects the same groups and differs only in the termination.

Since the algorithm in [20] identifies S_{g} under our assumptions, Algorithm 1 also identifies this set if it takes $|S_{\mathrm{g}}|$ steps. Furthermore, Algorithm 1 takes at most $|S_{\mathrm{g}}|$ steps since at step $|S_{\mathrm{g}}| + 1$, the residual vanishes and the while loop terminates. Finally, the fact that Algorithm 1 identifies S_{g} after $|S_{\mathrm{g}}|$ steps implies that if it terminates before $|S_{\mathrm{g}}|$ steps due to the λ stopping criterion, then the identified set at termination must be a subset of S_{g}. ■

The condition (11b) on the $D_{i,j}$ is a nontrivial requirement, both on the matrix $\boldsymbol{A}$ and on a basis set S_{g}, and it is difficult to verify in practice. This restriction is essentially an identification of classes of matrices for which we can guarantee that GOMP can be effective. In addition, since the set S_{g} defining a basis group is not unique for a given matrix, many choices of S_{g} may not satisfy (11b). One significant disadvantage of this analysis is that the condition (11b) cannot be checked before the set S_{g} has been identified.

For $\boldsymbol{A} \in \mathbb{R}^{d \times n}$, assuming $d \le n$ and that $\boldsymbol{A}$ has full rank, we require $|S_{\mathrm{g}}| = d$ in order to satisfy the relation (11a). However, d can be very large, and so in practice we will identify a set of features S with size $|S| < d$; this suggests that (11a) can only be satisfied approximately in this case. Following the analysis in [20], we can provide a robust version of the above theorem, showing the success of GOMP when the relation (11a) is satisfied only approximately.

We require a little more notation to proceed: With $\boldsymbol{A}$ and S_{g} as in Theorem 3.1, let $\boldsymbol{A}_{S_{\mathrm{g}}}$ be as defined in the proof of that theorem, i.e., the submatrix of $\boldsymbol{A}$ formed from the columns indexed by S_{g}. Now define the smallest eigenvalue of $\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}$:

(13) $$\bar{\lambda} := \lambda \left(\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}} \right) = \min_{\boldsymbol{v} \in \mathbb{R}^{|S_{\mathrm{g}}|} \backslash \{\boldsymbol{0}\}} \frac{\left\| \boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}} \boldsymbol{v} \right\|_2}{\left\| \boldsymbol{v} \right\|_2} > 0,$$

where the inequality is true since the columns of $\boldsymbol{A}_{S_{\mathrm{g}}}$ are a basis for range($\boldsymbol{A}$).

Theorem 3.2. *Let $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ have normalized columns $\boldsymbol{a}_i \in \mathbb{R}^d$, and let $S_{\mathrm{g}} \subset [n]$ be a basis set for $\boldsymbol{A}$ such that the columns of $\boldsymbol{A}$ satisfy*

(14a) $$\boldsymbol{a}_j = \sum_{i \in S_{\mathrm{g}}} D_{i,j} \boldsymbol{a}_i + \boldsymbol{n}_j,$$

and assume that each $\boldsymbol{n}_j \in \mathbb{R}^d$ for $j = [n] \backslash S_{\mathrm{g}}$ has independent and identically distributed components, each a normal random variable with mean 0 and variance σ^2. For any $\eta \in (0, 1/2)$, let the stopping tolerance ε in Algorithm 1 satisfy

$$
(14\mathrm{b}) \qquad \varepsilon > \frac{\sigma \sqrt{2 \cdot n \cdot d \log(2 \cdot n \cdot d / \eta)}}{1 - \bar{D}}.
$$

Assume that the $D_{i,j}$ satisfy (11b) and further that they satisfy

$$
(14\mathrm{c}) \qquad \min_{i \in S_{\mathrm{g}}} \sqrt{\sum_{j=1}^{n} D_{i,j}^2} > \frac{\varepsilon \sqrt{8}}{\bar{\lambda}}.
$$

Then, with probability at least $1 - 2\eta$, GOMP identifies the set S_{g} and computes a solution $\boldsymbol{B}$ satisfying

$$
\max_{i,j} |B_{i,j} - D_{i,j}| \leq \sigma \sqrt{\frac{2 \log(2|S_{\mathrm{g}}|/\eta)}{\bar{\lambda}}}.
$$

Proof. Let $\boldsymbol{D}$ be the matrix formed from the elements $D_{i,j}$, which is row-sparse, having nonzero entries only when $i \in S_{\mathrm{g}}$. As before, we seek a computational approximation $\boldsymbol{B}$ to the (unknown) $\boldsymbol{D}$. The model (14a) results in the following linear system to determine $\boldsymbol{B}$:

$$
\boldsymbol{AB} \approx \boldsymbol{A} = \boldsymbol{AD} + \boldsymbol{N},
$$

where $\boldsymbol{N}$ is a concatenation of the columns $\boldsymbol{n}_j$. (We define $\boldsymbol{n}_i = \boldsymbol{0}$ for $i \in S_{\mathrm{g}}$.) Following the proof of Theorem 3.1, the vectorization of the transpose of the above equation is

$$
\boldsymbol{Hg} = \boldsymbol{f} + \boldsymbol{n},
$$

where $\boldsymbol{n} = \mathrm{vec}\left(\boldsymbol{N}^T\right)$ and $\boldsymbol{f} = \mathrm{vec}\left(\boldsymbol{D}^T \boldsymbol{A}^T\right)$. The above equation holds when $\boldsymbol{g}$ has group sparsity defined by the set S_{g}. We seek to show that GOMP applied to this system, having knowledge only of $\boldsymbol{H}$ and $\boldsymbol{f} + \boldsymbol{n}$, can identify groups of $\boldsymbol{g}$ corresponding to the column indices S_{g} for $\boldsymbol{A}$.

The system above is simply a "noisy" version of GOMP, where $\boldsymbol{n}$ is a noise vector. We seek to apply Theorem 3 in [20] to show the conclusions. Assumptions (14) and (11b) are among the stipulations of this theorem. The remaining stipulation is

$$
(15) \qquad \lambda_{\min}\left(\boldsymbol{H}_{G_{\mathrm{g}}}^T \boldsymbol{H}_{G_{\mathrm{g}}}\right) \leq 1,
$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the symmetric matrix input. We can show this via applications of Kronecker product properties. First, we have

$$
\boldsymbol{H}_{G_{\mathrm{g}}}^T \boldsymbol{H}_{G_{\mathrm{g}}} = \left(\boldsymbol{A}_{S_{\mathrm{g}}} \otimes \boldsymbol{I}_n\right)^T \left(\boldsymbol{A}_{S_{\mathrm{g}}} \otimes \boldsymbol{I}_n\right) = \left(\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}\right) \otimes \left(\boldsymbol{I}_n \boldsymbol{I}_n\right).
$$

If λ_i, $i = 1, \ldots, |S_{\mathrm{g}}|$, are the eigenvalues of $\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}$, then they are also the eigenvalues of $\left(\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}\right) \otimes (\boldsymbol{I}_n \boldsymbol{I}_n)$, each having multiplicity n. Thus,

$$\lambda_{\min} \left(\boldsymbol{H}_{G_{\mathrm{g}}}^T \boldsymbol{H}_{G_{\mathrm{g}}} \right) = \lambda_{\min} \left(\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}} \right) = \bar{\lambda},$$

with $\bar{\lambda}$ defined in (13). The matrix $\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}}$ is positive-definite and has diagonal entries equal to 1 since the columns $\boldsymbol{a}_i$ are normalized. Thus,

$$|S_{\mathrm{g}}| = \mathrm{trace} \left(\boldsymbol{A}_{S_{\mathrm{g}}}^T \boldsymbol{A}_{S_{\mathrm{g}}} \right) = \sum_{i=1}^{|S_{\mathrm{g}}|} \lambda_i \geq |S_{\mathrm{g}}| \bar{\lambda},$$

so that $\bar{\lambda} \leq 1$, showing (15). Having shown this, we may apply Theorem 3 in [20], which yields the conclusion of the theorem. ∎

Note that (14a) can be satisfied with $|S_{\mathrm{g}}| < d$ even if $\boldsymbol{A}$ has full rank. When the columns of $\boldsymbol{A}$ are feature vectors that are outputs of scientific models, then the assumption that columns are perturbed by random noise is not realistic. However, the main purpose of Theorem 3.2 is to show that GOMP is robust to relaxations of (11a).

3.3. Theoretical discussion.
We describe briefly how the various methods we have investigated compare in terms of error or projection norm guarantees, number of columns, and asymptotic runtime to achieve those guarantees. We present certain results that we believe are relevant; the results we review are not meant to be comprehensive.

In Table 1, the relationship between ϵ and m for deterministic leverage score sampling assumes a power-law distribution, parameterized as $\ell_i = \frac{1}{i^{1+\eta}}$, on the leverage scores. The algorithms Cholesky, QR, LU, and GOMP are all greedy algorithms, and thus are analyzed using results from [1], which analyzes a general *greedy* column subset algorithm. For QR, we report the multiplicative error reported in [15], although the results from [1] apply as well.

We observe among the results we compile that leverage sampling has the most attractive theoretical error bounds because the bound is multiplicative and depends on the level of error, ϵ, which is a satisfactory metric in applications. The rest of the methods have the same error bound from [1], which is not directly related to the best rank-k approximation. In addition, QR has a multiplicative relationship to the best rank-k approximation, but the coefficient grows exponentially with respect to k.

In summary, the various algorithms we list in Table 1 can in theory be effective in choosing subsets. While the theoretical projection and error bounds do not allow us to conclude in practice which algorithm is superior, we aim to demonstrate in the results section that, for a variety of multifidelity problems, the GOMP strategy appears to be more effective than the rest.

4. Experimental results.

4.1. Methods compared.
We use a collection of methods from both the literature in previous work on multifidelity simulation approximation and in the more general CSSP literature. Previous work in multifidelity simulation approximation has primarily used pivoted Cholesky (chol) for subset selection, although the authors have alluded to the possibility of using other

Table 1

Theoretical comparison among the subset selection methods discussed here. Each method computes a size-m subset that can achieve a multiplicative-type bound relative to the optimal size-k subset, where m depends on k and an error tolerance ϵ. $\mathbf{C}$ is the computed subset matrix of rank m using the indicated algorithm, and $\mathbf{D}$ is the optimal subset matrix of rank k, where optimal means with respect to the projection error committed in the norm used in the "Theoretical error" column. ϵ is an error tolerance, and η is a weighting parameter for leverage scores.

Algorithm	Runtime	Theoretical error	$m = m(k, \epsilon)$
Leverage [25]	$\mathcal{O}(n^3)$	$\|\mathbf{A} - \mathbf{CC}^\dagger\mathbf{A}\|_F^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A_k}\|_F^2$	$m = \max\left(\left(\frac{2k}{\epsilon}\right)^{\frac{1}{1+\eta}}, \left(\frac{2k}{\eta\epsilon}\right)^{\frac{1}{\eta}}, k\right)$
Cholesky [1]	$\mathcal{O}(n^2 m)$	$\|\mathbf{CC}^\dagger\mathbf{A}\|_F^2 \geq (1-\epsilon)\|\mathbf{DD}^\dagger\mathbf{A}\|_F^2$	$m = \frac{16k}{\epsilon\sigma_{\min}(DD^\dagger)}$
QR [15]	$\mathcal{O}(n^2 m)$	$\|\mathbf{A} - \mathbf{CC}^\dagger\mathbf{A}\|_2 \leq \left(1 + \sqrt{n-k} \cdot 2^k\right)\|\mathbf{A} - \mathbf{A_k}\|_2$	$m = k$
LU [1]	$\mathcal{O}(n^2 m)$	$\|\mathbf{CC}^\dagger\mathbf{A}\|_F^2 \geq (1-\epsilon)\|\mathbf{DD}^\dagger\mathbf{A}\|_2^2$	$m = \frac{16k}{\epsilon\sigma_{\min}(DD^\dagger)}$
GOMP [1]	$\mathcal{O}(n^2 m^2)$	$\|\mathbf{CC}^\dagger\mathbf{A}\|_F^2 \geq (1-\epsilon)\|\mathbf{DD}^\dagger\mathbf{A}\|_F^2$	$m = \frac{16k}{\epsilon\sigma_{\min}(DD^\dagger)}$

Table 2

Summary of methods used in the empirical comparison for choosing resource allocation.

Abbreviation	Algorithm
rand	uniform random
lev	deterministic leverage score [25]
qr	pivoted QR [15]
chol	pivoted Cholesky [23]
lu	partially pivoted LU [14]
gomp	group orthogonal matching pursuit (section 3.1)
rank-k	best rank-k approximation [14]

pivoted decomposition methods, such as partially pivoted LU decomposition (lu) and pivoted QR decomposition (qr). All of these decomposition methods have a general approach of selecting the element with the largest residual norm after removing the contribution from previous selected elements, with the exception of LU, which does this but using the element with the maximum residual element magnitude (rather than the full norm). From the CSSP literature, we compare against the leverage score sample (lev) that makes use of the SVD of the Gram matrix to make the selection. Finally, we compare against the proposed method, grouped orthogonal matching pursuit (gomp), which was explained in detail in section 3.1. We also include results using a uniformly random selection (rand) and a best rank-k approximation (rank-k) as contextual benchmarks. Here, rank-k means the best approximating matrix of rank k measured in the ℓ^2 or Frobenius norm. The Eckart–Young–Mirsky theorem gives an explicit construction of this matrix as a truncated SVD of the full matrix. All methods used are summarized in Table 2.

We report the error as a squared ℓ_2 error normalized by the sum of squared ℓ_2 norms of the original data, $E = \frac{\sum_i \|\mathbf{X}_i - \tilde{\mathbf{X}}_i\|_2^2}{\sum_i \|\mathbf{X}_i\|_2^2}$, where $\mathbf{X}_i$ is the ith simulation and $\tilde{\mathbf{X}}_i$ is the proxy estimation for the ith simulation.

4.2. Burgers' equation. We use the viscous Burgers equation to evaluate the method in a well-studied and familiar setting. We introduce uncertain perturbations to the left boundary

condition and the viscosity parameter, similar to the setting studied in [38].

Specifically, we use

$$u_t + uu_x = \nu u_{xx}, \qquad x \in (-1, 1), \tag{16}$$
$$u(-1) = 1 + \delta, \qquad u(1) = -1, \tag{17}$$

where $\nu \sim U(0.1, 1)$ and $\delta \sim U(0, 0.1)$ are both uniform random variables in their respective ranges. This same problem with only the boundary perturbation (fixed viscosity) was studied in [38] because it is extremely sensitive to boundary perturbations [34]. We found that adding a second element of uncertainty around the viscosity parameter, ν, makes the problem more interesting from the perspective of attempting a linear approximation.

4.2.1. Setup. We sampled this parameter space using 400 total samples by taking 20 uniform grid samples within the boundary perturbation range, $(0, 0.1)$, and 20 uniform grid samples within the viscosity perturbation range, $(0.1, 1.0)$. For each parameter sample pair, we ran the low-fidelity and high-fidelity simulations for a set number of $t = 1.5 \times 10^6$ simulation steps, resulting in 400 different simulation results for the low-fidelity model and another 400 different simulation results for the high-fidelity model.

4.2.2. Analysis. The ultimate goal is to understand how each subset selection method performs using the low-fidelity for selection, while the error is computed with respect to the high-fidelity estimation. To test this, we use each method to select a subset and reconstruction weights using the low-fidelity dataset, and then we test the reconstruction error in both the low-fidelity samples and the high-fidelity samples.

We observed that all methods improved with each additional sample available, but certain methods do better overall and some outperform all others when the number of samples is extremely small. This latter case is important for our setting because we are in the setting where only a few samples can be computed in the high-fidelity model. We found that overall, deterministic leverage-score sampling performed the worst, even after a relatively large number of samples have been found. Note, however, that leverage sampling is frequently used in the context of very large datasets, and in this particular experiment we have access to only 400 samples. Thus, leverage sampling may be a good strategy for column subset selection if more data were available or when a larger number of subsamples are selected. In this experiment, random sampling appears to do better on average than deterministic leverage score sampling. However, when we look at the worst-case example from a random subset, we see that all other methods do better for small subsets. This last point is important when considering an allocation strategy of only a few high-cost simulations because while on average random does well it can do quite poorly in specific cases. The LU-based partial pivoting sampling approach did better than random but worse overall than the remaining methods. In all of our experiments, QR and Cholesky-based sampling performed nearly identically. This can be understood as follows: When acting on a symmetric positive-definite matrix like the Gramian, the QR and Cholesky pivoting strategies become nearly equivalent. Finally, the proposed GOMP-based sampling approach performs the best overall, and particularly for a very small subset setting. These results are summarized in Figure 1, which shows the reconstruction error in the low-fidelity model and the high-fidelity model against the exact solution.
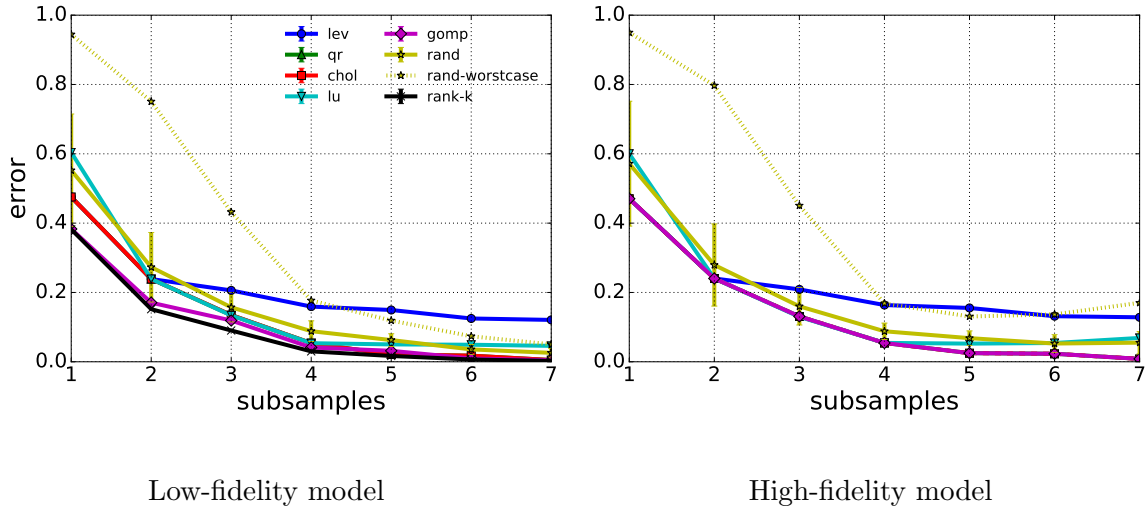
Low-fidelity model   High-fidelity model

Figure 1. *Burgers' equation: Reconstruction error among subset methods for solutions to Burgers' equation. Left: the reconstruction error of the rest of the low-fidelity simulation dataset when using the indicated subset chosen using various methods. Right: the reconstruction error but for the high-fidelity simulations with the subset chosen using the low-fidelity samples.*

The right-hand panel of Figure 1 (the high-fidelity error plot) only showcases errors from this multifidelity procedure, and not from, e.g., a best rank-k approximation on the high-fidelity ensemble of solutions. Our focus here is on the empirical study of the quality of high-fidelity subset selection using the low-fidelity data; under this focus, the shown plots are the most relevant and the error study of a high-fidelity rank-k approximation falls out of scope. Instead of subset selection quality, one can ask about how using the low-fidelity model in the reconstruction procedure affects the accuracy. In other words, this is a question about how the approximation to $\boldsymbol{A}^H$ in (2) would compare with an approximation using "high-fidelity" coefficients. In other words, it is a question concerning the accuracy of

$$\boldsymbol{A}^H \approx \boldsymbol{A}_S^H \left(\boldsymbol{A}_S^H\right)^\dagger \boldsymbol{A}^H \qquad \text{vs.} \qquad \boldsymbol{A}^H \approx \boldsymbol{A}_S^H \left(\boldsymbol{A}_S^L\right)^\dagger \boldsymbol{A}^L,$$

once S has been determined. The effect of this approximation is not the central goal of this paper, and so we leave this question for future investigation. However, we note that this question has been investigated in some recent work [16, 18]. The remaining figures in this manuscript therefore focus mainly on studies pertaining to the quality of the subset selection.

4.3. Double pendulum. We consider a classic double pendulum problem. A similar problem was also considered in [23], and we use the same setup and assumptions here. The problem is parameterized by the two pendulum angles, θ_1, θ_2, the lengths of the pendulums, ℓ_1, ℓ_2, and the mass of each pendulum, m_1, m_2, as well as the gravity coefficient, g. The high-fidelity model corresponds to the solution of equation (6.5) in [23] parameterized by (m_2, ℓ_2) using a strong stability preserving Runge–Kutta method. The low-fidelity model uses the same parameterization but a corresponding linear approximation; see equation (6.7) in [23] for details. In contrast to [23], we use a Euclidean inner product for the examples shown here.

4.3.1. Setup. For the high-fidelity model, we use a time-step $\Delta t = 10^{-2}$ until $T = 15$, resulting in a high-fidelity time-series vector of size 1501. The low-fidelity space uses $\Delta t = 0.25$ until $T = 15$, resulting in a 61-dimensional time-series vector. The uncertain parameter m_2 is sampled along the interval $[0.25, 0.75]$, and ℓ_2 is sampled from the range $[0.25, 4]$. We sample the low-fidelity model using 20 uniform grid samples along each parameter range, resulting in 400 simulations.

4.3.2. Analysis. For this example, we observed a much larger gap between the lowest and highest performers. This gap indicates that subset selection choice matters more for this problem, probably due to the more strongly nonlinear relationship exhibited here than in the Burgers example above.

The error observed in the low-fidelity model, shown in the left-hand side of Figure 2 on the left, demonstrates that the GOMP-based approach achieves superior performance in almost every subset size, until the number of samples becomes so large that the choice of subset selection does not matter. Using the low-fidelity subset choices in computing reconstructions for the high-fidelity model, shown in the right-hand side of Figure 2, achieves similar improvements. In the high-fidelity case, the model errors are much larger, making any gains more important.

A random subset selection does quite well on average in comparison to these deterministic approaches. However, we emphasize again that in the worst case a random subset can do quite poorly, as shown in the results, which makes it a risky strategy for simulation allocation. However, we note that rarely would making the allocation of a single high-fidelity simulation based purely on a random selection make sense, due to the risk of any specific subset performing quite poorly. While in expectation the random case performs well, any specific random subset may not, unless the subset size becomes large enough that the subset selection algorithm is no longer important. However, having arbitrarily large numbers of high-fidelity simulations is typically not possible for real-world situations.

4.4. Compressible flow simulation. Compressible flow simulations are used to study how fluids act while flowing in and around obstacles. Understanding these situations in detail is vital in many domains, especially in engineering scenarios where compressibility effects are important. We present here results from a 2D compressible flow simulation around a cylindrical object. The simulation finds a solution to the compressible Navier–Stokes equation, which for two dimensions can be written as

$$(18) \qquad \frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0,$$

where q is a vector of conserved variables, and $f = f(q, \nabla q)$ and $g = g(q, \nabla q)$ are the vectors of the fluxes. These can be rewritten as

$$(19) \qquad f = f_i - f_v, g = g_i - g_v,$$

where f_i, g_i are the inviscid fluxes as given in [17] and f_v, g_v are the corresponding viscous fluxes. The viscous fluxes take the form

$$(20) \qquad f_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} + kT_x \end{pmatrix}, \qquad g_v = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + kT_y \end{pmatrix},$$
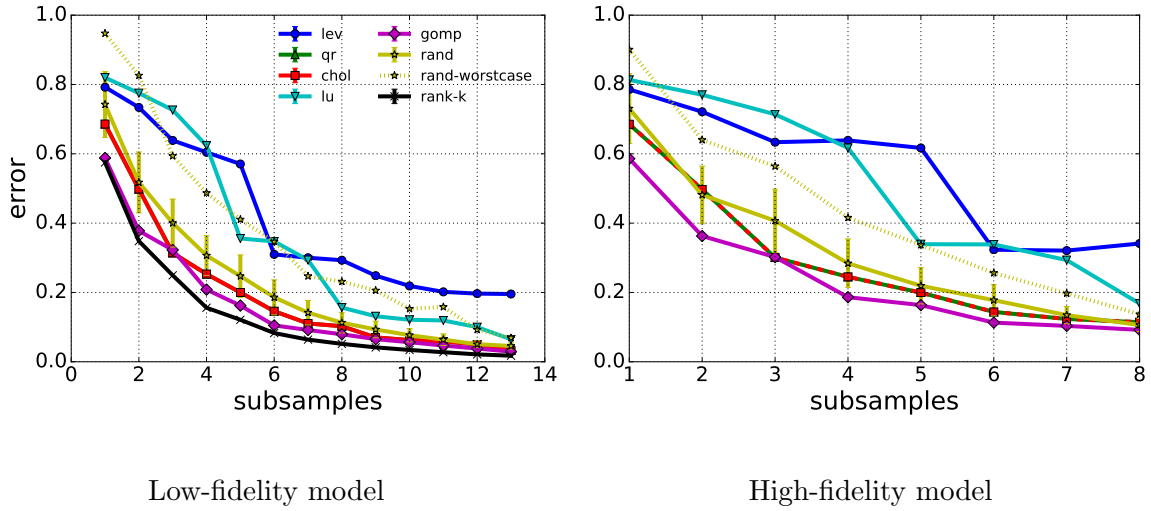
Low-fidelity model   High-fidelity model

Figure 2. *Double pendulum: Reconstruction error among subset methods for solutions to a double pendulum problem. Left: the reconstruction error of the rest of the low-fidelity simulation dataset when using the indicated subset chosen using various methods. Right: the reconstruction error but for the high-fidelity simulations with the subset chosen using the low-fidelity samples.*

where τ is the stress tensor,

$$(21) \qquad \tau_{xx} = 2\mu\left(u_x - \frac{u_x + u_y}{3}\right), \quad \tau_{yy} = 2\mu\left(v_y - \frac{u_x + u_y}{3}\right), \quad \tau_{xy} = \tau_{yx} = \mu(v_x + u_y),$$

with μ the dynamic viscosity and k the thermal conductivity. The corresponding solution is well known, and we refer the reader to classic texts for appropriate background, such as [22].

We are specifically interested in two uncertain parameters, the Reynolds number, Re, which relates the velocity of the fluid to the viscosity of the fluid, and the Mach number, Ma, which relates the speed of the fluid to the local speed of sound. We ran the simulations using the compressible flow module of the Nektar++ suite [8]. In the simulation, we use the following relationships between the Reynolds number and dynamic viscosity and between the Mach number and the *farfield* velocity:

$$(22) \qquad\qquad\qquad \mu = \frac{\rho_\infty u_\infty}{Re}, \qquad u_\infty = Ma\sqrt{\frac{\gamma p_\infty}{\rho_\infty}},$$

where $p_\infty, \rho_\infty, u_\infty$ denote the *farfield* pressure, density, and x-component of the velocity, respectively, and γ is the ratio of specific heats. An example of a single time point in one of the simulations is shown in the left-hand side of Figure 3.

4.4.1. Setup. We evaluated the Reynolds number, Re, and Mach number, Ma, parameters at regular intervals with five samples between 200 and 450, and six samples between 0.2 and 0.45, respectively. For a single parameter pair, we ran the simulation for 1×10^7 steps to achieve one unit of nondimensionalized simulation time, and for each time-step we recorded four values at 15 spatial locations in the immediate vicinity of the cylinder. The spatial locations are shown visually in Figure 3. We are primarily interested in the fluid flow simulation
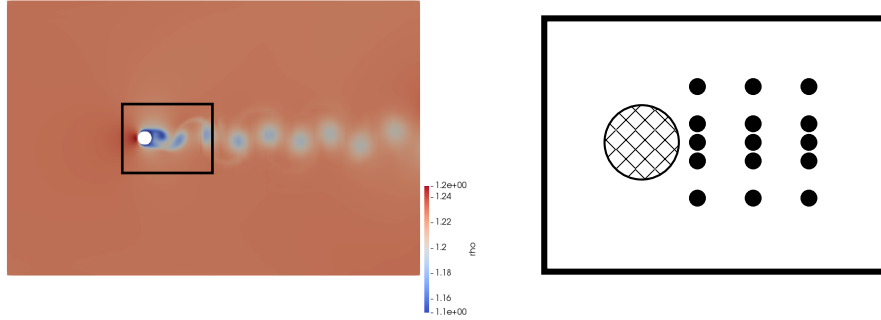
Figure 3. *Compressible flow simulation. Left: an example of one of the compressible flow simulations used in this experiment, showing the entire field of mass-density (ρ) values. The simulation models a fluid flowing past a cylinder; we are specifically interested in the mass-density and energy values that emerge in the immediate wake of the cylinder. Right: a zoomed-in diagram showing the location of the 15 sample points taken from each simulation for the analysis. These points were specifically positioned in the immediate wake of the cylinder. The location of this diagram is shown by the rectangle in the image on the left.*

after it transitions into a steady shedding regime, so we ignore any of the time-steps in the startup regime, which for this problem we observed to be at 30% of the steps, or 4×10^4 steps. This results in a feature vector with 6×10^4 time-steps and $15 \times 4 = 60$ values for each step.

Once the simulation transitions from the transient regime, all of these simulation sample points are periodic signals whose form depends nonlinearly on the parameters chosen. To align the signals, we detect the period length, find the cyclic peak, and translate all signals so that the first peak aligns. We then crop all signals to a single period size. The cropping was done to ensure a nice decay of singular values in the ensemble. A sample of a single positional point and variable but for various Reynolds and Mach numbers is shown in Figure 4 for reference. The corresponding Hilbert space is induced by the inner produce of these period length signals.

We ran both the low- and high-fidelity simulations over the specified parameter values in order to compute the proxy approximation error using the various subset selection methods.

4.4.2. Analysis. The results of the comparison in terms of how each of the subset selection methods perform are shown in Figure 5. Here we observe slightly differing results in the high-fidelity vs. low-fidelity errors. In the low-fidelity case, the methods perform similar to previous experiments. The GOMP, Cholesky, QR, and leverage methods all perform the same after a certain number of samples, and all perform better than random sampling. For the high-fidelity results, we observed that Cholesky and QR perform similarly to random, while GOMP performs the best. One of the primary differences between GOMP and QR is that GOMP performs selection based on residual correlation in the Hilbert space, while QR uses residual magnitude. These results indicate that the correlation becomes more indicative than the magnitude of high-fidelity sample importance for this type of simulation data; this is another reason to consider GOMP when choosing the subset selection technique.

4.5. Structure topology optimization. A structure topology optimization (STO) problem involves solving a nonconvex optimization problem for the optimal material placement to
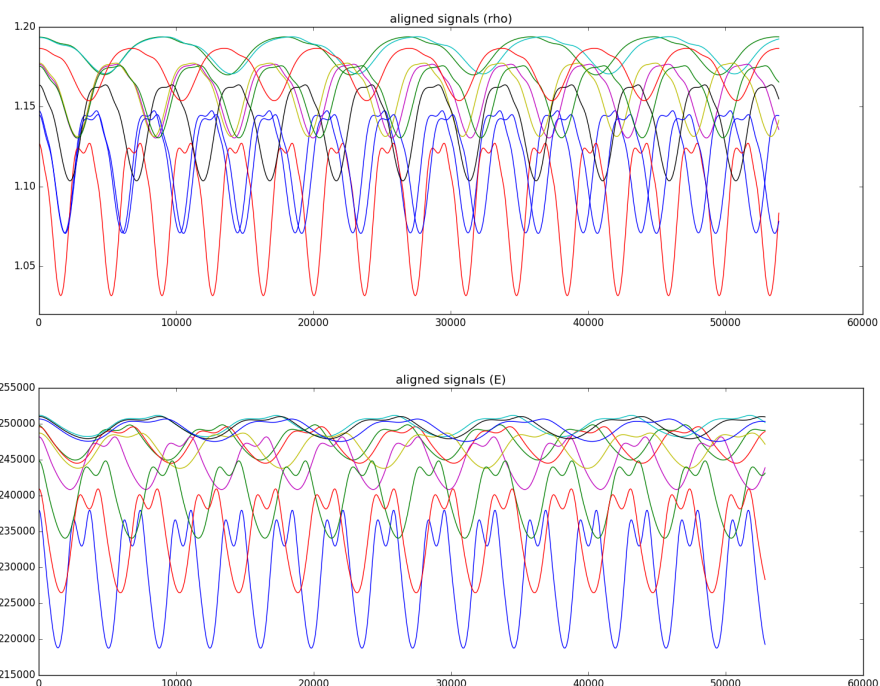
Figure 4. *Compressible flow simulation point samples over time. Top: samples of the mass-density field at a single positional point* (1.5, 0.0), *which is located in the immediate wake of the cylinder in Figure* 3. *The differing signal shapes are due to changes in the Reynolds and Mach numbers of the simulation; the figure shows* 10 *randomly sampled Reynolds–Mach pairs. Bottom: same as above but from the energy field.*

satisfy a physical property such as stiffness or maximum stress. STO has emerged as a powerful tool in designing various high performance structures, from medical implants and prosthetics to jet engine components [32, 30, 37]. The optimization shown here is a canonical STO problem in which the optimization finds the best material layout in the design space in order to maximize the stiffness (or, equivalently, minimizing the deflection or compliance) of the structure subject to a material volume constraint. The response of the structure for a set of loading and boundary conditions is typically computed/simulated via the FEM. The topology optimization yields a solution in the form of binary maps that indicate material placement.

The low- and high-fidelity models are binary map outputs; see the left-hand side of Figure 6. Our multifidelity procedure does not operate on the binary maps themselves but instead on a signed distance transform of the binary maps. To perform linear approximations, we compute signed distance transform (SDT) fields from binary maps and use these fields in the linear reconstruction. The SDT fields can be thresholded to recover binary maps; see the right-hand side of Figure 6 right for examples. All errors reported in this section are errors in the SDT fields. Figure 6 shows that thresholded SDT reconstructions can provide some macroscopic information about structures but can lose a substantial amount of small-scale structure.
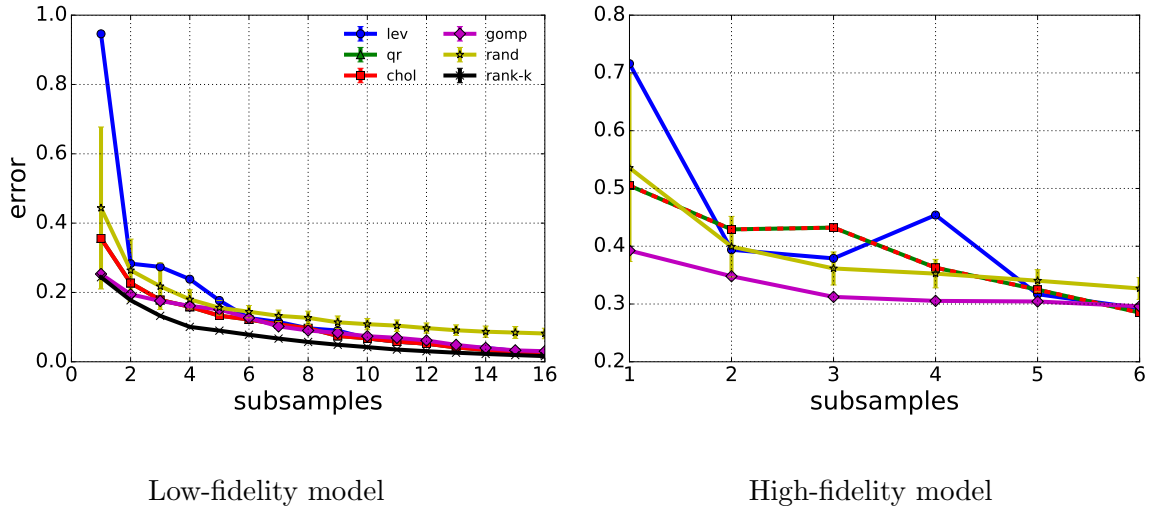
Low-fidelity model High-fidelity model

Figure 5. *Projection error among subset methods for solutions to compressible flow problems. Left: the reconstruction error of the rest of the low-fidelity simulation dataset when using the indicated subset chosen using various methods. Right: the reconstruction error for the high-fidelity simulations with the subset chosen using the low-fidelity samples.*

Additional details of the specific optimization and example implementations can be found in [2].

The problem considered here is parameterized by three variables, the vertical position of the loading, p, the loading angle, θ, and the filter size, ρ, which controls the scale of contiguous blocks in the material placement. This is summarized in Figure 7.

4.5.1. Setup. The topology optimization takes place on a 2D rectangular grid of size $n_{row} \times n_{col}$. In this experiment, the position variable, p, was restricted to $p \in [-0.5n_{row}, 0.5n_{row}]$ corresponding to the extent of the right boundary depicted in Figure 7. The angle parameter was restricted to $\theta \in [0, \pi]$ corresponding to the full range between pointing straight down to straight up. The filter parameter was restricted to $\rho \in [1.1z, 2.5z]$ corresponding to the scale of features allowed in the final simulation, where z is the appropriate ratio coefficient depending on the resolution s_x, s_y. We sampled the three parameter spaces in a uniformly random way until 1000 sample triples were found. Those triples were then used to run 1000 separate topology simulations. Low fidelity corresponds to the problem solved with $n_{row} = 40, n_{col} = 80, z = 1$, and high fidelity corresponds to the problem solved on a much larger region, $n_{row} = 80, n_{col} = 160, z = 2$.

4.5.2. Analysis. Our analysis takes into account both the low-fidelity and the high-fidelity spaces. Specifically, we analyze how well the subset selected by each method works to reconstruct the remaining low-fidelity samples. We also consider how the method reconstructs the high-fidelity samples using the same subset.

We observed that the GOMP approach performed the best for any subset size examined. The QR and Cholesky methods also performed quite well in the low-fidelity reconstruction error. The average of random method performed relatively well initially, but then all other
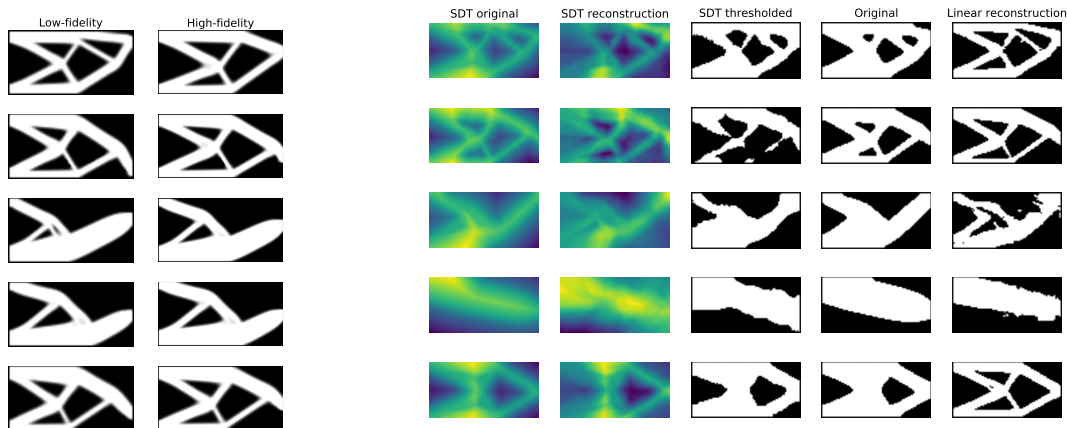
Figure 6. *STO. Left* 2 *columns: Binary maps output from both low- and high-fidelity solutions. Each row shows the low- and high-fidelity solutions using the same parameters; note the variety of changes in the solution just by moving from a lower resolution domain to a higher resolution domain. Right* 5 *columns: illustration of SDT fields resulting from binary maps output from the optimization process. The first column shows the SDT fields from the original binary maps (fourth column). The second column shows the SDT reconstruction, which is subsequently thresholded to obtain the binary structure (fifth column). For reference, the third column shows thresholding of the first-column SDT field.*
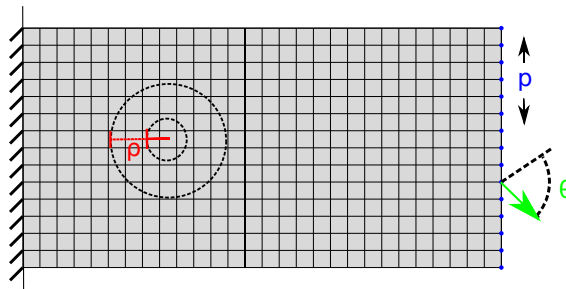
Figure 7. *STO diagram. The loading on the right edge is parameterized by a vertical position, p, and a loading angle, θ, and the filter size, ρ, which controls the length scale in the topology optimization.*

methods perform better at larger subset sizes.

The high-fidelity case was more different for this dataset than in the previous simulations examined. Specifically, we note that the GOMP maintains the position as the best performer; however, it also gains a considerable edge in the larger subset sizes. Surprisingly, Cholesky and QR perform well at first but then perform worse as the subset size grows. The average of random subsets does better than the LU and leverage approaches. The performance comparison among the different methods is summarized in Figure 8.

The poorer performance of QR for larger subset sizes, we hypothesize, is due to the larger
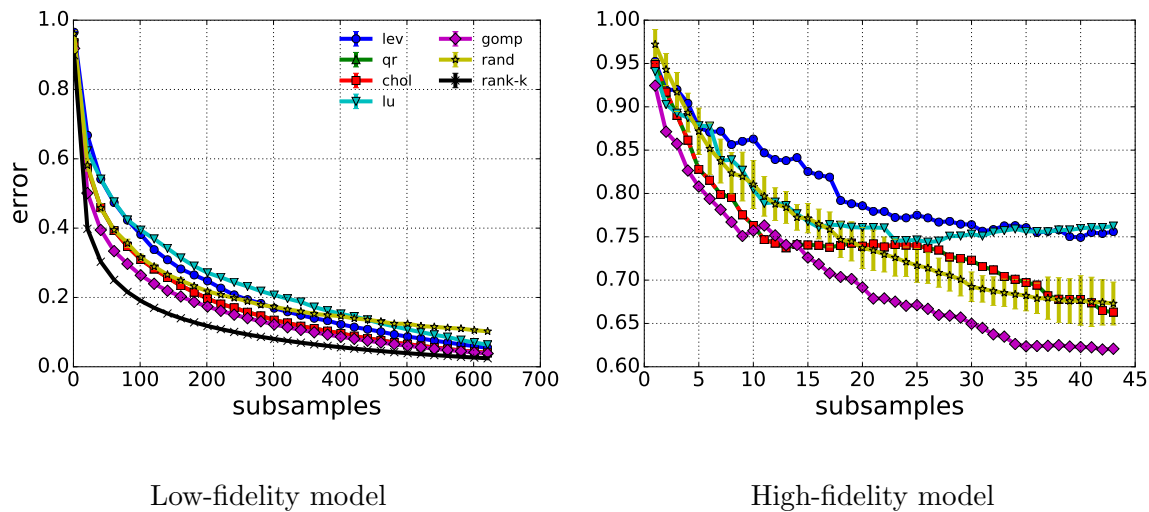
Low-fidelity model   High-fidelity model

Figure 8. *STO. Reconstruction error among subset methods. Left: the reconstruction error of the rest of the low-fidelity simulation dataset when using the indicated subset chosen using various methods. Right: the reconstruction error for the high-fidelity simulations with the subset chosen using the low-fidelity samples.*

set of differences between the low- and high-fidelity solutions in this dataset. Because of the nonconvex nature of the topology optimization problem, we observe some larger differences in the various solutions; an example is shown in Figure 6. We hypothesize that because GOMP bases the decision of using a subset element largely on overall correlation with the remaining data point, rather than simple residual error magnitude, it is able to capture the more important trends of the samples which carry over to the high-fidelity case.

Our previous analysis has focused on the subset size. A larger subset size for the same error ultimately translates into a higher runtime cost to obtain the same approximation error. To make this relationship more clear, we also compared the methods directly using simulation runtime and reconstruction error.

A popular alternative approach to generating proxy functions for applications in uncertainty analysis of simulations is to use a Gaussian process (GP) regression to estimate unseen simulations results (see, e.g., [19, 28, 27]). However, a GP approach relies on a large number of high-fidelity simulations from which to derive the proxy function. This is directly at odds with computationally constrained high-fidelity problems. We illustrate this point by including a GP proxy using the same, *small* number of samples used in the multifidelity approach.

We also consider an alternative neural-network-based proxy method proposed specifically for topology optimization in [33]. In [33], the authors proposed training a standard feedforward neural network, also known as a multilayered perceptron (MLP), as a proxy for the exact solution given the problem constraints. They first reduce the dimension of the dataset to an 80-dimensional space using PCA and then train the MLP to regress to the PCA weights given the problem constraints. We implemented their approach and got comparable results for the same number of solution designs used in their training (400). However, this approach also assumes the availability of sufficient data to train the model, which is not always the case in the high-fidelity models in the wild. To compare, we used both an MLP-based approach

with no PCA reduction and an MLP training on the PCA weights where the space was chosen to capture 90% of the singular value energy. The results are summarized in Figure 9. The left-hand side of Figure 9 compares all the methods on a log-scale. Because of the lack of training data, the MLP model performs quite poorly. The GP model performs much better initially than the MLP model on the fewer number of samples, but after reaching between 10–15 samples the two become quite similar in performance. Both the GP and the MLP models accrue considerable error because the limited samples do not sufficiently represent the high-fidelity space. The right-hand side of Figure 9 shows only the multifidelity methods on a linear scale. Note that it is difficult to directly compare results here with those in Figure 8 because this plot shows an average cost and runtime over 50 smaller datasets of size 500 each, which results in much smaller error overall. However, the relative performance of the methods is generally the same.

In contrast, the Gramian-weighted nonparametric proxy method introduced in [23] performs well using only a small number of high-fidelity samples because it makes direct use of the low-fidelity structure through the low-fidelity Gramian. In considering the choice in how the subset selection of the nonparametric proxy method is done, we found similar advantages in using a GOMP-based subset to the previous examples considered. Specifically, we note that given the same amount of high-fidelity simulation time we can improve the error considerably over previous methods. Our comparison in Figure 9 between our multifidelity methods (lev, qr, chol, lu, gomp) and surrogate methods (MLP, MLP-pca, GP) is not quite a comparison on an even playing field: The cost of training the multifidelity and surrogate methods is comparable, but evaluating the multifidelity approximations requires a low-fidelity evaluation, which can be more expensive than evaluation of a neural network or GP surrogate. However, this example illustrates that in our situation neural networks and GPs cannot be trained on only high-fidelity data with reasonable cost. In scenarios when the high-fidelity model is expensive, it is thus more difficult to accurately train surrogate models, and it is in this regime when the multifidelity approximation can be useful.

There are some significant differences between the observed reconstruction error performance among the different subset methods. For the topology optimization problem, these differences can be seen easily by viewing the subsets of solutions chosen and considering the differences. Figure 10 shows the first 15 subsets chosen by each of the methods in order of selection. Note that QR and Cholesky chose the same subsets for this dataset, and for brevity we only show QR. All of the methods select a similar solution initially, one of the solid beam structure solutions. This is consistent with our understanding of each of the methods, as the leverage method is selecting the sample with the most statistical leverage, QR, Cholesky, and LU are selecting the sample with the largest residual, and GOMP is selecting the sample with the largest correlation with all the samples. Note how QR selects a sample with some slant, and this probably results in a larger magnitude image, while GOMP selects a beam with a straight orientation, and this is probably higher correlated with more samples. The methods start to differentiate after a few samples: Statistical leverage continues to select samples with similar large beam structures because those solutions also have a large statistical leverage score (leverage scores do not take into account prior samples like the other methods). The order of the QR and Cholesky selections is interesting because, as the first sample was slanted one way, the residual error indicated that a slant the other way was the next best accord-
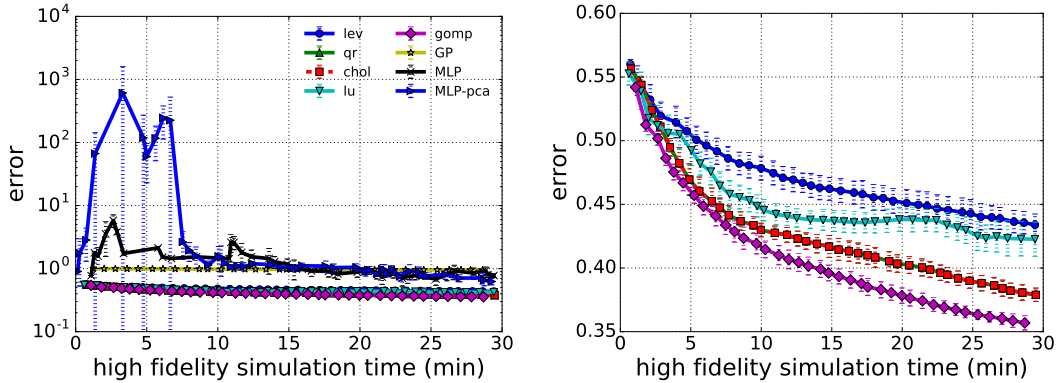
Figure 9. *STO. Average reconstruction error (vertical axis) vs. high-fidelity simulation runtime (horizontal axis) among subset methods over an ensemble of 50 random datasets containing 500 samples each; the standard deviation of the ensemble error is shown in dashed error bars around each point. The "error" is the relative ℓ^2 vector error averaged over all samples. From left to right, each point indicates the number of high-fidelity simulations used to obtain the approximation, and the vertical and horizontal location is the average error and average time to simulation, respectively, over the 50 dataset realizations for that number of simulations. Left: a log-scale comparison including a GP, an MLP, and an MLP training on PCA projections (MLP-pca) similar to [33]. Right: a linear-scale view of the same results but only showing the multifidelity techniques.*

ing to the magnitude criteria. After the slanted beams, other solutions with large beams in them as well as small lattices were chosen, probably because these also resulted in large residual magnitudes. The GOMP approach, in contrast, selected two large beam solutions, and then selected a solution made entirely of lattices, probably because that distribution of material more highly correlated with the remaining residuals. We hypothesize that because GOMP uses residual correlation instead residual magnitude, it is able to do better in the high-fidelity space even if there are some differences between low- and high-fidelity solutions for the topology optimization problem.

5. Conclusion. We have investigated a novel approach to determine a design of experiments for high-fidelity simulations in a multifidelity framework. Low-rank multifidelity techniques require a step that identifies an appropriate allocation of scarce computational resources in a parametric high-fidelity model. Existing methods accomplish this allocation by using a particular greedy technique to select a small subset from a large candidate. For the purposes of performing this multifidelity approximation, we explore and compare many existing alternative subset selection techniques that are popular in statistics and machine learning. We find that a GOMP algorithm can consistently yield superior results on challenging datasets.

We provide first-principles motivation of the problem and the choice of GOMP, and we present analysis suggesting that GOMP can be effective for this task. Our numerical results show substantial improvement in the multifidelity error per simulation cost. The GOMP algorithm was compared to previous allocation strategies as well as existing column subset selection problem solutions on several simulation datasets. This work thus investigates an unexplored connection between the multifidelity allocation problem and the classic subset selection problem in the machine learning and data mining domains. Our analysis provides
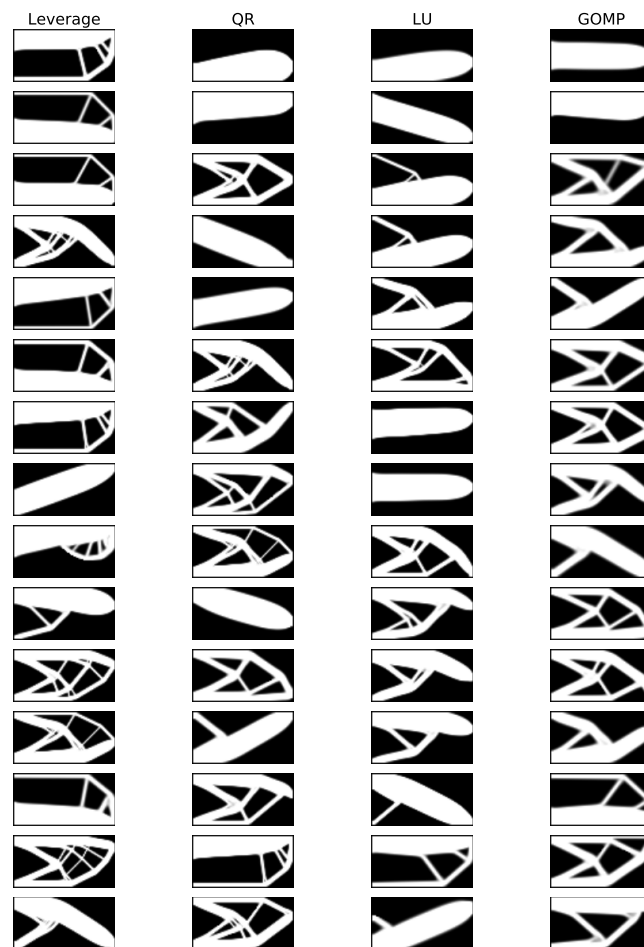
Figure 10. *STO samples: the samples chosen (in order) by each algorithm. Because Cholesky and QR select the same samples, we only show the results for QR in the interest of space. Note that all methods select samples with a large primary beam first and then become quite different afterwards. The differences are caused by how the methods make the next selection; for example, GOMP uses max correlation with residual, while QR/Cholesky use max residual magnitude.*

a first step to understanding the apparent superiority of the GOMP strategy, but a more rigorous analysis requires further investigation.

REFERENCES

[1] J. ALTSCHULER, A. BHASKARA, A. ROSTAMIZADEH, M. ZADIMOGHADDAM, T. FU, AND V. MIRROKNI, *Greedy column subset selection: New bounds and distributed algorithms*, in Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016.

[2] E. ANDREASSEN, A. CLAUSEN, M. SCHEVENELS, B. S. LAZAROV, AND O. SIGMUND, *Efficient topology optimization in MATLAB using 88 lines of code*, Struct. Multidiscip. Optim., 43 (2011), pp. 1–16.

[3] F. BACH, *Sharp analysis of low-rank kernel matrix approximations*, in Proceedings of the 30th International Conference on Learning Theory (ICML), 2013, pp. 185–209.

[4] C. BOUTSIDIS, P. DRINEAS, AND M. MAGDON-ISMAIL, *Near-optimal column-based matrix reconstruction*, SIAM J. Comput., 43 (2014), pp. 687–717, https://doi.org/10.1137/12086755X.

[5] C. BOUTSIDIS, M. W. MAHONEY, AND P. DRINEAS, *An improved approximation algorithm for the column subset selection problem*, in Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 968–977, https://doi.org/10.1137/1.9781611973068.105.

[6] S. BOYD, *Alternating direction method of multipliers*, in Proceedings of the 51st IEEE Conference on Decision and Control, Vol. 3, 2011, pp. 1–44.

[7] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.

[8] C. D. CANTWELL, D. MOXEY, A. COMERFORD, A. BOLIS, G. ROCCO, G. MENGALDO, D. DE GRAZIA, S. YAKOVLEV, J.-E. LOMBARD, D. EKELSCHOT, B. JORDI, H. ZU, Y. MOHAMEID, C. ESKILSSON, B. NELSON, P. VOS, C. BIOTTO, AND R. M. KIRBY, *Nektar++: An open-source spectral/hp element framework*, Comput. Phys. Commun., 192 (2015), pp. 205–219.

[9] T. F. CHAN AND P. C. HANSEN, *Some applications of the rank revealing QR factorization*, SIAM J. Sci. Comput., 13 (1992), pp. 727–741, https://doi.org/10.1137/0913043.

[10] H. CHENG, Z. GIMBUTAS, P. G. MARTINSSON, AND V. ROKHLIN, *On the compression of low rank matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1389–1404, https://doi.org/10.1137/030602678.

[11] M. B. COHEN AND C. MUSCO, *Ridge Leverage Scores for Low-Rank Approximation*, preprint, https://arxiv.org/abs/1511.07263v1, 2015.

[12] W. DENG, W. YIN, AND Y. ZHANG, *Group Sparse Optimization by Alternating Direction Method*, Proc. SPIE 8858, SPIE, Bellingham, WA, 2013, https://doi.org/10.1117/12.2024410.

[13] G. GOLUB, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Vol. 3, John Hopkins University Press, Baltimore, MD, 2012.

[15] M. GU AND S. C. EISENSTAT, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869, https://doi.org/10.1137/0917055.

[16] J. HAMPTON, H. FAIRBANKS, A. NARAYAN, AND A. DOOSTAN, *Parametric/Stochastic Model Reduction: Low-Rank Representation, Non-intrusive Bi-fidelity Approximation, and Convergence Analysis*, preprint, http://arxiv.org/abs/1709.03661, 2017.

[17] G. E. KARNIADAKIS AND S. J. SHERWIN, *Spectral/hp Element Methods for CFD*, Oxford University Press, New York, 2005.

[18] V. KESHAVARZZADEH, R. M. KIRBY, AND A. NARAYAN, *Convergence acceleration for time-dependent parametric multifidelity models*, SIAM J. Numer. Anal., submitted.

[19] L. LE GRATIET, *Multi-fidelity Gaussian Process Regression for Computer Experiments*, Ph.D. thesis, Université Paris-Diderot-Paris VII, Paris, 2013.

[20] C. LOZANO, G. SWIRSZCZ, AND N. ABE, *Group orthogonal matching pursuit for variable selection and prediction*, in Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS), 2009, pp. 1150–1158.

[21] M. W. MAHONEY AND P. DRINEAS, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 697–702, https://doi.org/10.1073/pnas.0803205106.

[22] B. R. MUNSON, D. F. YOUNG, AND T. H. OKIISHI, *Fundamentals of Fluid Mechanics*, 3rd ed., John Wiley & Sons, New York, 1990.

[23] A. NARAYAN, C. GITTELSON, AND D. XIU, *A stochastic collocation algorithm with multifidelity models*,

SIAM J. Sci. Comput., 36 (2014), pp. A495–A521, https://doi.org/10.1137/130929461.

[24] L. W.-T. NG AND M. ELDRED, *Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation*, in 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics, Reston, VA, 2012, AIAA 2012-1852, https://doi.org/10.2514/6.2012-1852.

[25] D. PAPAILIOPOULOS, A. KYRILLIDIS, AND C. BOUTSIDIS, *Provable deterministic leverage score sampling*, in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2014, pp. 997–1006, https://doi.org/10.1145/2623330.2623698.

[26] B. PEHERSTORFER, K. WILLCOX, AND M. GUNZBURGER, *Optimal model management for multifidelity Monte Carlo estimation*, SIAM J. Sci. Comput., 38 (2016), pp. A3163–A3194, https://doi.org/10.1137/15M1046472.

[27] P. PERDIKARIS, M. RAISSI, A. DAMIANOU, N. LAWRENCE, AND G. KARNIADAKIS, *Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling*, Proc. A, 473 (2017), 20160751.

[28] P. PERDIKARIS, D. VENTURI, J. O. ROYSET, AND G. E. KARNIADAKIS, *Multi-fidelity modelling via recursive co-kriging and Gaussian-Markov random fields*, Proc. A, 471 (2015), 20150018, https://doi.org/10.1098/rspa.2015.0018.

[29] D. J. PERRY AND R. T. WHITAKER, *Augmented leverage score sampling with bounds*, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Berlin, Heidelberg, 2016, pp. 543–558.

[30] T. A. REIST, J. ANDRYSEK, AND W. L. CLEGHORN, *Topology optimization of an injection moldable prosthetic knee joint*, Comput. Aided Design Appl., 7 (2010), pp. 247–256.

[31] Y. SHITOV, *Column Subset Selection Is NP-Complete*, preprint, https://arxiv.org/abs/1701.02764, 2017.

[32] A. SUTRADHAR, G. H. PAULINO, M. J. MILLER, AND T. H. NGUYEN, *Topological optimization for designing patient-specific large craniofacial segmental bone replacements*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 13222–13227.

[33] E. ULU, R. ZHANG, AND L. B. KARA, *A data-driven investigation and estimation of optimal topologies under variable loading configurations*, Comput. Methods Biomech. Biomed. Eng. Imaging Vis., 4 (2016), pp. 61–72.

[34] D. XIU AND G. E. KARNIADAKIS, *Supersensitivity due to uncertain boundary conditions*, Internat. J. Numer. Methods Engrg., 61 (2004), pp. 2114–2138.

[35] Y. YANG AND H. ZOU, *A fast unified algorithm for solving group-lasso penalize learning problems*, Stat. Comput., 25 (2015), pp. 1129–1141.

[36] M. YUAN AND Y. LIN, *Model selection and estimation in regression with grouped variables*, J. R. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), pp. 49–67.

[37] J.-H. ZHU, W.-H. ZHANG, AND L. XIA, *Topology optimization in aircraft and aerospace structures design*, Arch. Comput. Methods Eng., 23 (2016), pp. 595–622.

[38] X. ZHU, A. NARAYAN, AND D. XIU, *Computational aspects of stochastic collocation with multifidelity models*, SIAM/ASA J. Uncertain. Quantif., 2 (2014), pp. 444–463, https://doi.org/10.1137/130949154.