SIAM/ASA J. Uncertainty Quantification
Vol. 12, No. 1, pp. 157–181

Stacking Designs: Designing Multifidelity Computer Experiments with Target Predictive Accuracy[*]

Chih-Li Sung[†], Yi (Irene) Ji[‡], Simon Mak[‡], Wenjia Wang[§], and Tao Tang[¶]

Abstract. In an era where scientific experiments can be very costly, multifidelity emulators provide a useful tool for cost-efficient predictive scientific computing. For scientific applications, the experimenter is often limited by a tight computational budget, and thus wishes to (i) maximize predictive power of the multifidelity emulator via a careful design of experiments, and (ii) ensure this model achieves a desired error tolerance with some notion of confidence. Existing design methods, however, do not jointly tackle objectives (i) and (ii). We propose a novel stacking design approach that addresses both goals. A multilevel reproducing kernel Hilbert space (RKHS) interpolator is first introduced to build the emulator, under which our stacking design provides a sequential approach for designing multifidelity runs such that a desired prediction error of $\epsilon > 0$ is met under regularity assumptions. We then prove a novel cost complexity theorem that, under this multilevel interpolator, establishes a bound on the computation cost (for training data simulation) needed to achieve a prediction bound of ϵ. This result provides novel insights on conditions under which the proposed multifidelity approach improves upon a conventional RKHS interpolator which relies on a single fidelity level. Finally, we demonstrate the effectiveness of stacking designs in a suite of simulation experiments and an application to finite element analysis.

Key words. computer experiments, experimental design, finite element analysis, Gaussian process modeling, multilevel modeling, uncertainty quantification

MSC code. 00A20

DOI. 10.1137/22M1532007

1. Introduction.

With recent developments in scientific computing and mathematical modeling, computer experiments have now become an essential tool in solving many scientific and engineering problems. These experiments, which typically solve complex mathematical models representing reality, are useful in applications which are prohibitively expensive or infeasible for direct experimentation. Such virtual experiments have now been successfully applied in a broad range of problems, from nuclear physics [13] to rocket design [30]. As

[*] Received by the editors October 31, 2022; accepted for publication (in revised form) November 27, 2023; published electronically March 11, 2024.

https://doi.org/10.1137/22M1532007

Funding: The work of the first author was supported by NSF DMS-2113407. The work of the second and third authors were supported by NSF CSSI Frameworks 2004571, NSF DMS-2210729, NSF DMS-2316012, and DE-SC0024477. The work of the fifth author was supported by the Statistical and Applied Mathematical Sciences Institute.

[†] Department of Statistics and Probability, Michigan State University, East Lansing, MI 48824 USA (sungchih@msu.edu).

[‡] Department of Statistical Science, Duke University, Durham, NC 27710 USA (yi.ji@duke.edu, sm769@duke.edu).

[§] Data Science and Analytics Thrust, Information Hub, Hong Kong University of Science and Technology uangzhou, Hong Kong (wenjiawang@ust.hk).

[¶] Department of Mathematics, Duke University, Durham, NC 27710 USA (tao.tang250@duke.edu).

the science becomes more sophisticated, however, such simulations can become prohibitively costly for parameter space exploration. A popular solution is *emulation* [39], which makes use of a carefully designed training set from the simulator to build an efficient predictive model that emulates the expensive computer code. Popular emulator models include the Gaussian process (GP) model or reproducing kernel Hilbert space (RKHS) interpolators [38, 9, 17, 18], neural networks [46, 33], and polynomial chaos methods [55], all of which have demonstrated successes in various areas of application.

For full-scale complex scientific systems, however, it is often the case that the training data needed to train an accurate emulator model can be prohibitively expensive to generate from the simulator. One way to address this is via *multifidelity emulation,* which supplements the costly high-fidelity (or high-accuracy) simulation dataset with less expensive lower-fidelity (or lower-accuracy) approximations for fitting the emulator model. The idea is that, by leveraging useful information from cheaper lower-fidelity simulations to enhance predictions for the high-fidelity model, an accurate multifidelity emulator can be trained with fewer high-fidelity runs and thus lower simulation costs. The usefulness of this multifidelity emulation framework has led to much work in recent years. A popular framework is the Kennedy–O'Hagan (KO) model [23], which models a sequence of computer simulations from lowest to highest fidelity using a sequence of GP models linked by a linear autoregressive framework. Recent developments on the KO model include [36, 24, 26, 25, 35, 20] (among many others), which investigated modeling strategies for efficient posterior prediction and Bayesian uncertainty quantification. For multifidelity simulators controlled by a single mesh parameter (e.g., mesh density in finite element analysis), [49] proposed a nonstationary GP model which leverages data at different mesh densities to predict the highest-fidelity simulation at the finest mesh. This model has been further developed for conglomerate multifidelity emulation [21] and graphical multifidelity emulation [20].

Despite this body of work, there remains important unresolved needs, particularly on the *design* of such multifidelity experiments for cost-efficient emulation. In modern scientific computing problems, the experimenter is often limited by a tight computational budget dictated by available computing resources for a project; see, e.g., [6]. Given such constraints, one thus wishes to (i) *maximize the* predictive power of the multifidelity emulator via a careful design of experiments, and (ii) ensure the resulting emulator achieves a desired prediction error bound with some notion of *confidence.* Despite its importance, however, there is little work to our knowledge on design methods for multifidelity emulators which jointly tackles objectives (i) and (ii). Recent works have addressed objective (i) in various ways. For instance, [25] proposes one-sample-at-a-time and batch sequential designs using fast cross-validation techniques for multifidelity simulations. [43] suggest a sequential design that maximizes uncertainty reduction while considering the simulation cost. [11] introduce a sequential design strategy that maximizes the *mutual information* for the next sampling location and minimizes the theoretical error bound to select the most effective fidelity level. However, these works do not fully address objective (ii). Hence, we propose a novel *stacking design* framework which aims to jointly address (i) and (ii).

In what follows, we let $f_l(x)$ denote the scalar (deterministic) simulation output of the computer code with input parameters $x \in \Omega \subseteq \mathbb{R}^d$ and at fidelity level l. In finite element analysis (FEA), this fidelity level may reflect the underlying mesh density of the numerical simulator. We further suppose that, as fidelity level l increases, the simulated output $f_l(x)$

approaches a limiting solution of $f_\infty(x)$, the desired "exact" solution of the simulator. In practice, this exact solution often cannot be simulated numerically; for example, the limiting solution for FEA (i.e., at an infinitely dense mesh density) often cannot be computed numerically. Our proposed method makes use of a multilevel RKHS interpolator, which leverages simulated data of multiple fidelities to train an emulator for predicting the desired limiting solution $f_\infty(x)$.

With the multilevel interpolator, the proposed stacking design aims to carefully choose the multifidelity experimental runs, to target a desired prediction bound of $\epsilon > 0$ between the multilevel interpolator and the desired solution $f_\infty(\cdot)$. This is achieved by iterating the following sampling steps. First, for a fixed number of sampled fidelity levels L, we show that the multilevel interpolator provides an easy-to-evaluate expression for allocating sample sizes over each fidelity level. After performing runs with these sample sizes in a space-filling fashion, we then derive a novel stopping rule for deciding, under regularity conditions, whether the number of levels L is sufficiently large for achieving the desired error bound ϵ. If L is not large enough, we then increment L and repeat the above design step. The resulting batch sequential design procedure creates a "stacking" effect, where design points are stacked on at each fidelity level until the stopping rule is satisfied, at which point the desired prediction bound ϵ is satisfied under regularity conditions. This stacking behavior is visualized in the bottom of Figure 1, where we see the stacking of sample sizes at each fidelity stage as the batch sequential design progresses. The proposed stacking designs are inspired by a similar sequential approach for multilevel Monte Carlo (MLMC; see [15]), which aims to provide cost-efficient error control for MLMC simulations. We then demonstrate the effectiveness of

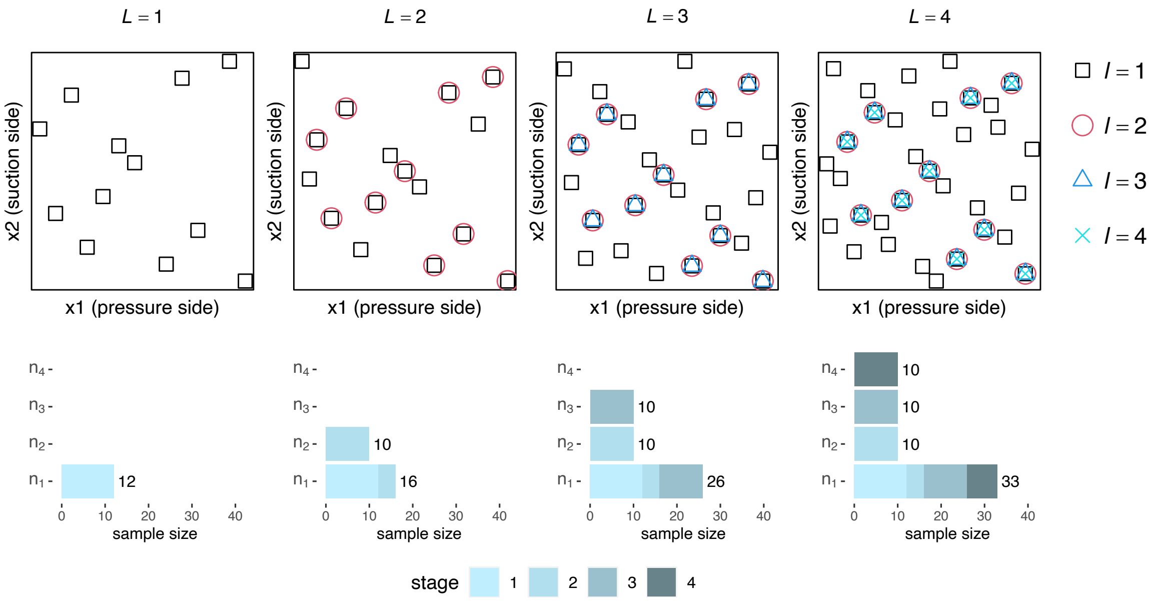

Figure 1. *Visualizing the stacking behavior of the proposed stacking designs for a jet engine turbine blade application investigated later. (Top) The proposed stacking designs at $L = 4$ fidelity levels, over four batch sequential design stages (from left to right). (Bottom) The corresponding sample sizes at the $L = 4$ fidelity levels, over four batch sequential design stages (from left to right).*

the proposed stacking designs (in terms of cost efficiency and error guarantees) in a suite of simulation experiments and an application to FEA.

A key novelty of our work is a new *cost complexity theorem* which, under the multilevel interpolator, establishes a bound on the required computation cost (for training data generation) needed to ensure the desired prediction bound ϵ. Such a result sheds useful insight on when multifidelity emulation may be most (or least) effective given a computation cost budget. As a corollary, we then show that the presented multifidelity approach yields provably improved predictions over a conventional single-fidelity RKHS interpolator, under intuitive conditions on the error decay and cost complexity of the multifidelity simulator. These results are again inspired by existing work in MLMC for characterizing the cost complexity of multilevel Monte Carlo estimators. To our knowledge, there has been little work on extending such results for characterizing cost complexity of multifidelity computer experiments; we aim to address this gap here.

The paper is organized as follows. Section 2 introduces the multilevel RKHS interpolator. Section 3 presents the proposed stacking designs in several steps. Section 4 discusses a cost complexity theorem for the multilevel interpolator. Section 5 investigates the effectiveness of stacking designs via a suite of simulation studies and an application to FEA. Section 6 concludes the paper. Proofs and code for reproducing numerical results are provided in supplemental materials (supplementary_materials.pdf [local/web 252KB]).

2. Multilevel RKHS interpolator.

We first introduce a multilevel interpolator, which we leverage later for our stacking designs. Again, let $f_l(x)$ denote the scalar simulation output of the computer code, with input parameters $x \in \Omega \subseteq \mathbb{R}^d$ and at fidelity level l. In what follows, we assume that L distinct fidelity levels have been sampled for training data, where a larger fidelity level indicates a higher fidelity (or higher accuracy) simulator with higher computational costs per run.

The goal then is to construct an efficient surrogate model, with uncertainty quantification, for the highest-fidelity (and thus most expensive) simulation code $f_L(x)$. Suppose, for the lth fidelity level, simulations are performed at the design points $\mathcal{X}_l = \{x_i^{[l]}\}_{i=1}^{n_l}$, where the sample size n_l varies for different fidelity levels l. This yields the simulation outputs $f_l|_{\mathcal{X}_l} = (f_l(x))_{x \in \mathcal{X}_l}$, where $f|_{\mathcal{X}}$ denotes the vector of outputs for $f(x)$ at design points $x \in \mathcal{X}$. For this multilevel emulator, we further assume that the designs $\mathcal{X}_l$ are sequentially nested, i.e.,

$$\mathcal{X}_L \subseteq \mathcal{X}_{L-1} \subseteq \cdots \subseteq \mathcal{X}_1 \subseteq \Omega. \tag{2.1}$$

In other words, design points run for a higher-fidelity simulator will be contained within the design points run for a lower-fidelity simulator.

With this, the multilevel interpolator is constructed as follows. Note that the high-fidelity response surface f_L can be decomposed as $f_L = \sum_{l=1}^{L}(f_l - f_{l-1})$, where $f_0 \equiv 0$, and $(f_l - f_{l-1})$ can be viewed as the *discrepancy* between the $(l-1)$th and lth code, capturing *refinements* in the response surface as fidelity increases. Suppose that $\Phi_l(x, x')$ is a positive-definite kernel function used for interpolating the refinement $(f_l - f_{l-1})|_{\mathcal{X}_l}$, then the RKHS interpolator of $(f_l - f_{l-1})|_{\mathcal{X}_l}$ has the simple form [52, 18]

$$\mathcal{P}_l(x) = \sum_{i=1}^{n_l} \alpha_i^l \Phi_l(x, x_i^{[l]}),$$

where $(\alpha^l_1,\dots,\alpha^l_{n_i})^T = \mathbf{\Phi}_l^{-1}\mathbf{z}_l$ with $\mathbf{\Phi}_l = [\Phi_l(x_i^{[l]}, x_j^{[l]})]_{i,j=1,\dots,n_l}$ and $\mathbf{z}_l := (f_l - f_{l-1})|_{\mathcal{X}_l}$. Then, a multilevel RKHS interpolator is introduced as follows:

$$\hat{f}_L(x) = \sum_{l=1}^{L} \mathcal{P}_l(x). \tag{2.2}$$

The resulting interpolator $\hat{f}_L(x)$ has some similarities to the cokriging models [23, 26, 25], which also assume nested designs but models instead the function $(f_l - \rho_{l-1} f_{l-1})$ via a GP prior using a Bayesian framework, where ρ_{l-1} is an unknown parameter. A potential alternative model is the nonstationary GP proposed in [49]. One advantage of the interpolator (2.2), as discussed in [11], is that it explicitly models for the bias between the exact simulation solution and its surrogate model, thus allowing for quantifying the emulation accuracy in theory, which is developed in the next section. For further insight on the comparison between GP and RKHS interpolators, we refer readers to the works of [50, 27, 40, 22], and potential extensions to GP emulators are discussed in section 6.

In the following, we make use of the Matérn kernel $\Phi_l(x_i, x_j) = \phi_{\nu_l}(\|\Theta_l^{-1}(x_i - x_j)\|_2)$, where

$$\phi_{\nu_l}(r) = \frac{2^{1-\nu_l}}{\Gamma(\nu_l)} \left(r\sqrt{2\nu_l}\right)^{\nu_l} B_{\nu_l}\left(r\sqrt{2\nu_l}\right). \tag{2.3}$$

Here, $\nu_l > 0$ is a smoothness parameter which controls differentiability of the interpolator, Θ_l is a diagonal $d \times d$ matrix of length-scale parameters, B_{ν_l} is the modified Bessel function of the second kind, and Γ is the gamma function. Matérn kernels are widely used in the computer experiment [39, 17] and spatial statistics [42] literature, and we will show later that such a kernel choice yields useful insights into characterizing the cost complexity performance of stacking designs.

Of course, the hyperparameters Θ_l and ν_l are unknown in practice and need to be estimated. These hyperparameters can be chosen to minimize the cross-validation error [18]. Although cross-validation methods are typically expensive to implement, the leave-one-out cross-validation (LOOCV) error of RKHS interpolators can be expressed in a closed form [18], which makes the computation more efficient. In particular, the LOOCV error of the interpolator $\mathcal{P}_l(x)$ is

$$\frac{1}{n_l}\|\mathbf{\Lambda}_l^{-1}\mathbf{\Phi}_l^{-1}\mathbf{z}_l\|_2^2, \tag{2.4}$$

where $\mathbf{\Lambda}_l$ is a diagonal matrix with the element $(\mathbf{\Lambda}_l)_{j,j} = (\mathbf{\Phi}_l^{-1})_{j,j}$, and the hyperparameters Θ_l and ν_l can be chosen by minimizing (2.4).

3. Stacking designs.

With the multilevel RKHS interpolator in hand, we now introduce the proposed stacking designs. We first define some notation. Let $\mathcal{N}_\Phi(\Omega)$ be the RKHS associated with a kernel Φ, and let $\|g\|_{\mathcal{N}_\Phi(\Omega)}$ denote its RKHS norm for a function $g \in \mathcal{N}_\Phi(\Omega)$. In the following, we consider the refinement $(f_l - f_{l-1})$ to live on the RKHS $\mathcal{N}_{\Phi_l}(\Omega)$.

Suppose now, for each fidelity level l, there corresponds a measure of fidelity $\xi_l > 0$ quantifying how close the simulated response surface $f_l(\cdot)$ is to the *exact solution,* which we denote as $f_\infty(\cdot)$. As $l \to \infty$, it is intuitive that $\xi_l \to 0$, meaning we approach the limiting solution as fidelity level increases. However, the exact solution $f_\infty(\cdot)$ typically cannot be computed numerically and must be approximated. In the case of the finite element method (FEM), which is widely used for computer experiments, one such parameter for ξ_l is the mesh size: a smaller mesh size ξ_l results in higher mesh density and thus a more accurate simulator, at the cost of higher computation.

We now investigate the error in approximating the desired exact solution $f_\infty(x)$ with the multifidelity interpolator $\hat{f}_L(x)$ in (2.2), which can be decomposed as

$$|f_\infty(x) - \hat{f}_L(x)| \le \underbrace{|f_\infty(x) - f_L(x)|}_{\text{simulation error}} + \underbrace{|f_L(x) - \hat{f}_L(x)|}_{\text{emulation error}}. \tag{3.1}$$

The first term corresponds to the *simulation error,* which measures the discrepancy between the simulated solution $f_L(x)$ at fidelity level L and the true/exact function $f_\infty(x)$. This error can be reduced by increasing L, the fidelity level L of the simulator, or, equivalently, by reducing ξ_l. The second term represents *emulation error*—the error for the multilevel interpolator given limited evaluations of the simulators. This error can be reduced by increasing n_L, the sample size at the fidelity level L, or increase $n_{L-1}, \ldots, n_1$, the sample sizes at lower–fidelity levels.

With this, the proposed stacking designs provide a batch sequential design scheme that aims to achieve the desired prediction accuracy of $\epsilon > 0$, i.e., $\|f_L - \hat{f}_L\| \le \epsilon$. Here, $\|\cdot\|$ may be the L_2 or L_∞ norm. This can be achieved by making both the simulation and emulation errors smaller than $\epsilon/2$, i.e.,

$$\|f_\infty - f_L\| \le \epsilon/2 \quad \text{and} \quad \|f_L - \hat{f}_L\| \le \epsilon/2. \tag{3.2}$$

We thus propose our stacking designs in two parts. In section 3.1, we first present a sample size determination approach which bounds the emulation error $\|f_L - \hat{f}_L\|$ via closed-form expressions for sample sizes at each fidelity level. In section 3.2, we then present a useful stopping rule on the maximum fidelity L such that the simulation error bound on $\|f_\infty - f_L\|$ is satisfied. We then discuss a sequential algorithm for stacking designs in section 3.3, and prove a novel complexity theorem that supports its performance in section 4.

3.1. Emulation error control.

Consider first the emulation error $\|f_L - \hat{f}_L\|$. We first present a useful bound on this error for the multilevel interpolator (proof in Supplementary Material SM1).

Proposition 3.1. *Suppose Ω is bounded and convex, $(f_l - f_{l-1}) \in \mathcal{N}_{\Phi_l}(\Omega)$. Assume that there exist constants $C_1, C_2, C_3, C_4, \nu_{\min}, \nu_{\max} > 0$ such that $C_1 \le \|\Theta_l^{-1}\|_2 \le C_2$, $C_3 \le \|\Theta_l\|_2 \le C_4$, $\nu_{\min} \le \nu_l \le \nu_{\max}$ for all $l = 1, \ldots, L$. Then one can bound the prediction error of $\hat{f}_L(x)$ as*

$$|f_L(x) - \hat{f}_L(x)| \le c_0 \sum_{l=1}^{L} \|\Theta_l^{-1}\|_2^{\nu_l} h_{\mathcal{X}_l}^{\nu_l} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \tag{3.3}$$

for some constant $c_0 > 0$. Here, $h_{\mathcal{X}_l}$ is the fill distance [52] of the design $\mathcal{X}_l$, i.e., $h_{\mathcal{X}_l} =$ $\sup_{x \in \Omega} \min_{x_u \in \mathcal{X}_l} \|x - x_u\|_2$.

This proposition nicely decomposes the prediction error of the multilevel interplator $\hat{f}_L$ into three distinct components at each fidelity level $l=1,\ldots,L$. The first term, $\|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}$, captures the size of the refinement with respect to its corresponding RKHS norm. This is quite intuitive, since a larger norm of the refinement $f_l - f_{l-1}$ is expected to induce greater error. The second term, $h_{\mathcal{X}_l}^{\nu_l}$, measures the quality of the design $\mathcal{X}_l$ in terms of how well it fills the design space Ω. Note that a smaller fill distance $h_{\mathcal{X}_l}$ suggests that there are fewer "gaps" between design points [28], which in turn should reduce prediction error. The third term, $\|\Theta_l^{-1}\|_2^{\nu_l}$, captures the magnitude of the length scales Θ_l. These three components provide the basis for the stacking sequential design method presented next.

We now wish to minimize the error bound in (3.3) under a total budget on computational resources, to yield easy-to-evaluate expressions for determining sample sizes n_l at each fidelity level $l=1,\ldots,L$. Let C_l be the computational cost (e.g., in CPU hours) for a single run of the simulator at fidelity level l. Note that, since higher-fidelity simulators are more computationally intensive, this implies that $0<C_1<C_2<\cdots<C_L$.

From the experimental design perspective, an appealing design criterion for interpolation is *quasi-uniformity* [52], which ensures design points are uniformly placed over the design space Ω. Specifically, denote $q_{\mathcal{X}} = \min_{i\neq j}\|x_i - x_j\|/2$, and a design $\mathcal{X}=\{x_i\}_{i=1}^n$ satisfying $h_{\mathcal{X}}/q_{\mathcal{X}} \leq c$, for some constant c, is called a quasi-uniform design. Such a design satisfies the fill distance bound [52, 32] $h_{\mathcal{X}} \leq c_1 n^{-1/d}$ for some constant $c_1 > 0$, where n is the number of design points in $\mathcal{X}$. Quasi-uniformity has been widely studied in the literature [14], and there are a variety of designs which enjoy this property [54]. We thus restrict the designs $\mathcal{X}_l$ to be quasi-uniform for $l=1,\ldots,L$. The construction of such designs is discussed later in section 3.3. Under this restriction, the error bound in (3.3) reduces to

$$
\begin{aligned}
|f_L(x) - \hat{f}_L(x)| &\leq c_0 \sum_{l=1}^{L} c_1^{\nu_l} \|\Theta_l^{-1}\|_2^{\nu_l} n_l^{-\nu_l/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \\
&\leq c_0 c_1^* \sum_{l=1}^{L} \|\Theta_l^{-1}\|_2^{\nu_l} n_l^{-\nu_{\min}/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}, \qquad (3.4)
\end{aligned}
$$

where $c_1^* = \max_{l=1,\ldots,L} c_1^{\nu_l}$ and $\nu_{\min} = \min_{l=1,\ldots,L} \nu_l$.

Consider now the sample size determination problem, where we wish to minimize the error bound in (3.4) under the constraint of the total computational budget, $\sum_{l=1}^{L} n_l C_l$. This can be done by the method of Lagrange multipliers, which aims to find the saddle point of the Lagrangian function

$$
\sum_{l=1}^{L} \left(\|\Theta_l^{-1}\|_2^{\nu_l} n_l^{-\nu_{\min}/d} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} + \lambda n_l C_l \right),
$$

where $\lambda > 0$ is the Lagrange multiplier. With some algebraic manipulations (by setting the gradient of the above function to zero), one can show that the optimal sample size for n_l is

$$
n_l = \mu r_l, \quad \text{where } r_l = \left(\frac{\|\Theta_l^{-1}\|^{\nu_l}}{C_l} \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \right)^{d/(\nu_{\min}+d)}, \qquad (3.5)
$$

for some constant $\mu > 0$. To ensure that n_l is an integer, in our later implementation, we set it to the floor value of μr_l, i.e., $n_l = \lfloor \mu r_l \rfloor$.

The closed-form expression (3.5) reveals several useful insights for sample size determination in multifidelity experiments. First, with all things being equal, we see that (3.5) allocates greater sample size n_l for simulations with lower costs C_l (i.e., lower-fidelity simulations), which is intuitive. Second, note that (3.5) assigns a greater sample size n_l for fidelity levels where the refinement $f_l - f_{l-1}$ is more complex, whether that be in terms of a smaller length scale Θ_l or a larger RKHS norm. In particular, note that the RKHS norm of $f_l - f_{l-1}$ captures *dissimilarities* of the simulators from fidelity level $l-1$ to level l. Thus, by minimizing (3.5), our approach naturally factors in this dissimilarity information for optimal sample size allocation.

There is still a free constant μ, which we can set to achieve the desired emulation error $\|f_L - \hat{f}_L\| < \epsilon/2$. This parameter can be optimized as follows. By Theorem 11.14 of [52], an alternative pointwise error bound of $\hat{f}_L(x)$ is

$$|f_L(x) - \hat{f}_L(x)| \leq \sum_{l=1}^{L} \sigma_l(x) \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}, \tag{3.6}$$

where $\sigma_l(x)$ is the so-called power function of the form

$$\sigma_l^2(x) = \Phi_l(x,x) - \Phi_l(x,\mathcal{X}_l)\mathbf{\Phi}_l^{-1}\Phi_l(x,\mathcal{X}_l)^T, \tag{3.7}$$

where $\Phi_l(x,\mathcal{X}_l) = \{\Phi_l(x,y)\}_{y\in\mathcal{X}_l}$. In contrast from the error bound in Proposition 3.1, the bound (3.6) does not depend on any constants, which allows for the following development. By the triangle inequality, $\|f_L - \hat{f}_L\|$ can be bounded by

$$\|f_L - \hat{f}_L\| \leq \sum_{l=1}^{L} \|\sigma_l\| \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}. \tag{3.8}$$

To ensure the bound in (3.8) is less than the desired error tolerance $\epsilon/2$, one can set the constant μ by solving the optimization problem

$$\mu^* = \underset{\mu>0}{\operatorname{argmin}} \left| \frac{\epsilon}{2} - \sum_{l=1}^{L} \|\sigma_l\| \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \right| \text{ s.t. } \sum_{l=1}^{L} \|\sigma_l\| \|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)} \leq \frac{\epsilon}{2}. \tag{3.9}$$

This optimization ensures that the error bound is equal (or close) to $\epsilon/2$, while the constraint ensures that it remains below $\epsilon/2$. Here, the dependency of the objective function on μ is via the term $\|\sigma_l\|$. This is because $\sigma_l(x)$ depends on the sample size n_l and subsequently μ (recall $n_l = \lfloor \mu r_l \rfloor$), since the power function (3.7) depends on the design points $\mathcal{X}_l$. To optimize for μ^* in (3.9), $\|\sigma_l\|$ can be approximated numerically via Monte Carlo integration [5] for the L_2 norm and grid search optimization for the L_∞ norm. While one can alternatively set a fixed total cost budget (i.e., $\sum_{l=1}^L n_l C_l$) for determining μ, we will focus here on achieving a desired predictive accuracy threshold ϵ.

Finally, to optimize μ^*, we also require knowledge of the RKHS norm $\|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}$. Of course, the exact norm is unknown in implementation since it depends on f_l and f_{l-1}. One can, however, approximate this term via its RKHS interpolator, which can be shown to be equal to $(\mathbf{z}_l^T \mathbf{\Phi}_l^{-1} \mathbf{z}_l)^{1/2}$ [52]. It should be noted that prior to optimization (i.e., before obtaining $\mathcal{X}_l$), we require information about the data $\mathbf{z}_l$ and the matrix $\mathbf{\Phi}_l$ in order to approximate the RKHS norm. These can be initially obtained via simulations on a pilot sample (we shall call this $\mathcal{X}_0$ later), then adaptively updated after collecting additional data from our stacking design (see section 3.3). Similarly, the kernel parameters Θ_l and ν_l, which are also required for the optimization (3.9) and for obtaining the sample size as in (3.5), can also be adaptively estimated using the pilot sample $\mathcal{X}_0$ via cross validation as mentioned in section 2. With these plug-in estimates within (3.9), the desired μ^* can then be efficiently obtained via simple dichotomous search [1].

We note that the above sample size determination approach has several key distinctions from that in [11]. First, our proposed sample sizes are determined by the desired prediction accuracy ϵ via (3.9), while theirs are controlled by a fixed total cost budget. This becomes important in the following subsection, where we leverage such sample sizes within a sequential stacking approach for controlling prediction error. Second, our approach makes use of an initial pilot sample to estimate the RKHS norm and other required parameters in (3.5), which lends well to our later sequential design procedure. The approach in [11], by contrast, makes use of misspecified kernel functions, in particular, the rates proved in [51].

3.2. Simulation error control via stacking.

Consider next the simulation error $\|f_\infty - f_L\|$ in (3.2), which concerns the numerical error of the simulator at fidelity level L. For many numerical simulators, this error can be bounded as

$$|f_\infty(x) - f_l(x)| \leq c_1(x)\xi_l^\alpha \quad \text{for all } x \in \Omega \tag{3.10}$$

for some positive constants α and $c_1(x)$ depending on x. Recall that ξ_l is a fidelity parameter which quantifies how close $f_l(\cdot)$ is to the exact solution $f_\infty(\cdot)$; the smaller ξ_l is, the higher the fidelity of the simulator $f_l(\cdot)$. Equation (3.10) thus assumes that the simulation error is decaying polynomially in the fidelity parameter ξ_l. In the case of FEM with ξ_l taken as the finite element mesh size, it is well known that the bound (3.10) holds under regularity conditions on the underlying solution (see, e.g., [4, 49]). Similarly polynomial decay rates have also been shown in a broad range of numerical simulators, e.g., for elliptical PDEs [19] and large-eddy simulations in fluid mechanics [45]. We will thus assume the error bound in (3.10), and leverage this for controlling simulation error via the stacking designs presented next.

Suppose the fidelity parameters $\{\xi_1, \xi_2, \ldots\}$ follow a geometric sequence for increasing fidelity levels, i.e., $\xi_l = \xi_0 T^{-l}, l \in \mathbb{N}^+$ for some integer $T \geq 2$. In the setting of FEM, where ξ_l measures finite element mesh size, $\xi_1, \xi_2, \ldots$ correspond to mesh sizes for increasing mesh refinements. The use of such a geometric sequence is motivated by MLMC [15, 16], which makes use of a similar sequence for time discretization of stochastic differential equations.

We now wish to ensure the simulation error satisfies the desired bound of $\|f_\infty - f_L\| \leq \epsilon/2$. Let us first assume a slightly stronger condition (compared to (3.10)),

$$f_\infty(x) - f_L(x) = c_1(x)\xi_L^\alpha + O(\xi_L^{\alpha+1}), \tag{3.11}$$

where $O(\xi_L^{\alpha+1})$ denotes a leading error term on the order of ξ_L to the $(\alpha+1)$th power. This assumption is common in the scientific computing literature, where it is referred to as the α-order *discretization error* to the mesh spacing parameter ξ_L [2, 34]. Using the above sequence, $\xi_l = \xi_0 T^{-l}$, and (3.11), it follows that

$$f_\infty(x) - f_{L-1}(x) = c_1(x)\xi_L^\alpha T^\alpha + O(\xi_L^{\alpha+1}). \tag{3.12}$$

By multiplying (3.11) by T^α and subtracting (3.12), we get

$$f_\infty(x) = \frac{1}{T^\alpha - 1}(T^\alpha f_L(x) - f_{L-1}(x)) + O(\xi_L^{\alpha+1}).$$

It thus follows that

$$f_\infty(x) - f_L(x) = \frac{f_L(x) - f_{L-1}(x)}{T^\alpha - 1}, \tag{3.13}$$

where terms of order $\xi_L^{\alpha+1}$ and higher are neglected. Of course, the numerator $(f_L(x) - f_{L-1}(x))$ is unknown in implementation; we can, however, estimate it via its RKHS interpolator, namely, $\mathcal{P}_L(x)$. Combining everything together, the following criterion then serves as a check for whether the desired simulation error bound $\|f_\infty - f_L\| \leq \epsilon/2$ is met:

$$\frac{\|\mathcal{P}_L(x)\|}{T^\alpha - 1} \leq \frac{\epsilon}{2}. \tag{3.14}$$

The above procedure extends a similar argument made in [15] for bounding approximation error in MLMC.

From (3.14), the rate parameter α plays an important role for ensuring $\|f_\infty - f_L\| \leq \epsilon/2$. One way to set this parameter, as suggested in [49], is to infer α via the known error bound (3.10) from numerical analysis. For example, in FEM, if the interest lies in the integration of $f_\infty(\cdot)$ over a specific region, then $\alpha = 2$ is suggested [49]. If such prior information is not available from numerical analysis, then one could instead estimate this rate from data. In particular, using the error expansion (3.11), we can write

$$\begin{aligned}
f_l(x) &= f_\infty(x) + c_1(x)T^{(L-l)\alpha}\xi_L^\alpha + O(\xi_L^{\alpha+1}),\\
f_{l-1}(x) &= f_\infty(x) + c_1(x)T^{(L-l+1)\alpha}\xi_L^\alpha + O(\xi_L^{\alpha+1}),\\
f_{l-2}(x) &= f_\infty(x) + c_1(x)T^{(L-l+2)\alpha}\xi_L^\alpha + O(\xi_L^{\alpha+1}).
\end{aligned}$$

Neglecting terms of $\xi_L^{\alpha+1}$ and higher, and subtracting f_{l-1} from f_l and f_{l-2} from f_{l-1} yield

$$\begin{aligned}
f_l(x) - f_{l-1}(x) &= c_1(x)T^{(L-l)\alpha}\xi_L^\alpha(1 - T^\alpha), \quad \text{and}\\
f_{l-1}(x) - f_{l-2}(x) &= c_1(x)T^{(L-l)\alpha}\xi_L^\alpha T^\alpha(1 - T^\alpha),
\end{aligned}$$

which gives

$$\alpha = \frac{\log\left(\frac{f_{l-1}(x) - f_{l-2}(x)}{f_l(x) - f_{l-1}(x)}\right)}{\log T} \quad \text{for} \quad l = 3, \dots, L.$$

Thus, the parameter α can be estimated by the average value evaluated by the data,

$$\hat{\alpha} = \frac{1}{L-2}\sum_{l=3}^{L}\sum_{x\in\mathcal{X}_l}\frac{\log\left(\left|\frac{f_{l-1}(x)-f_{l-2}(x)}{f_l(x)-f_{l-1}(x)}\right|\right)}{n_l \log T}, \tag{3.15}$$

where the absolute value is used to ensure a positive value within the logarithm.

It is worth noting that the above developments have direct analogues in the scientific computing literature, in terms of the *grid convergence index* method and the assessment of *observed order of accuracy.* Such methods are commonly used to quantify discretization uncertainty in numerical models governed by partial differential equations. The foundation of these developments lies in a reinterpretation of Richardson's extrapolation procedure [37], as employed in the above derivations; further details can be found in Chapter 8 of [34]. A comparative analysis between GCI and a Bayesian alternative for discretization uncertainty quantification [49] is discussed in [2].

3.3. Stacking design algorithm. We now combine the two error control approaches in sections 3.1 and 3.2 into a sequential algorithm for stacking designs. From previous developments, there are two key properties that the design points in $\mathcal{X}_l$, $l = 1,\ldots,L$, should satisfy: (i) they should be nested over fidelity levels, i.e., $\mathcal{X}_L \subseteq \cdots \subseteq \mathcal{X}_1$, and (ii) for each fidelity level l, the design $\mathcal{X}_l$ should satisfy the quasi-uniformity property discussed in section 3.1. One way to satisfy both properties is first choose a quasi-uniform sequence $\{z_i\}_{i=1}^{\infty}$ on the domain Ω, then construct the multilevel designs as $\mathcal{X}_l = \{z_i\}_{i=1}^{n_l}$, $l = 1,\ldots,L$. Although there are methods for constructing quasi-uniform sequences (see, e.g., the low-dispersion sequences [56, 3], such approaches typically have inflexible sample sizes or are not easily adaptable in a batch sequential manner (as needed for stacking designs). In our later implementation, we made use of Sobol' sequences [41], which have been used empirically as sequential quasi-uniform designs (see, e.g., [47]). While it is unclear whether such designs achieve the optimal rates required for quasi-uniformity [44, 54], it appears to yield good empirical performance in our numerical studies.

We now describe the proposed stacking design algorithm in detail. The algorithm begins by selecting an initial fidelity level $L = 1$, then choosing an initial pilot design $\mathcal{X}_0$ of size n_0. Here, we suggest the initial sample size n_0 to be $5d \sim 10d$. We then iterate the following steps:

1. Run the simulator at fidelity level L and observe $f_L(x)$ at the pilot design $\mathcal{X}_0$. Estimate the hyperparameters $\{\Theta_l\}_{l=1}^{L}$ and $\{\nu_l\}_{l=1}^{L}$ by minimizing the LOOCV error (2.4), and the RKHS norms $\{\|f_l - f_{l-1}\|_{\mathcal{N}_{\Phi_l}(\Omega)}\}_{l=1}^{L}$ using the approach in section 3.1.
2. Using these estimated parameters, compute the optimal sample sizes n_l via (3.5) and (3.9) for the current fidelity levels $l = 1,\ldots,L$. With this, construct designs $\mathcal{X}_l$ (with sample size n_l) to satisfy a nested structure, i.e., $\mathcal{X}_0 \subseteq \mathcal{X}_L \subseteq \mathcal{X}_{L-1} \subseteq \cdots \subseteq \mathcal{X}_1$. Run the simulators at these design points at their respective fidelity levels.
3. If the number of fidelity levels $L \geq 3$, estimate the rate parameter α in (3.10) via regression (see section 3.2). Using this estimate, test convergence via the stopping rule (3.14).
4. If convergence is not satisfied, iterate $L \leftarrow L+1$ and repeat the above three steps. Otherwise, stop the batch sequential design and return the multilevel RKHS interpolator (2.2).

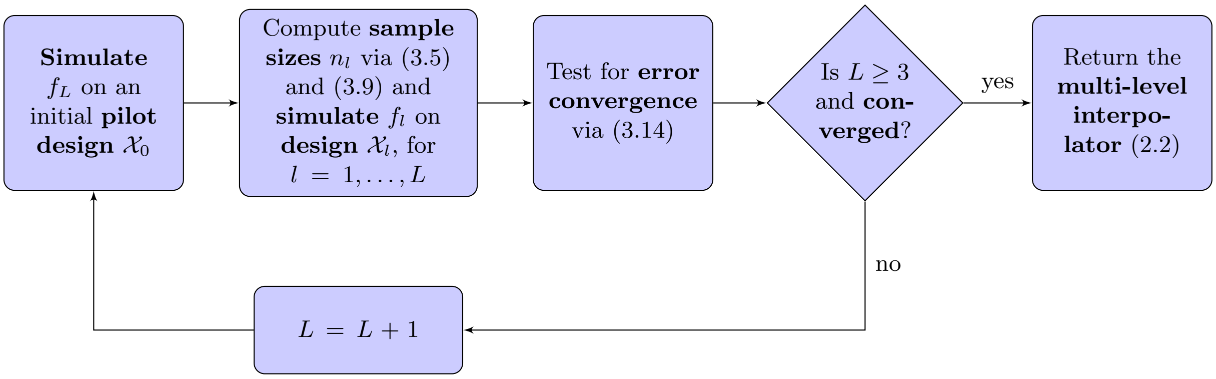
Simulate f_L on an initial **pilot design** $\mathcal{X}_0$
Compute **sample sizes** n_l via (3.5) and (3.9) and **simulate** f_l on **design** $\mathcal{X}_l$, for $l = 1, \ldots, L$
Test for **error convergence** via (3.14)
Is $L \geq 3$ and **converged**?
yes
Return the **multi-level interpolator** (2.2)
no
$L = L + 1$

Figure 2. *Visualizing the sequential workflow for the proposed stacking design algorithm.*

Note that the computed sample sizes from step 2 aim to control the emulation error $|f_L(x) - \hat{f}_L(x)|$ (see section 3.1), and the stopping rule in step 3 aims to control the simulation error $|f_\infty(x) - f_L(x)|$ (see section 3.2). The workflow of this batch sequential design algorithm is visualized in Figure 2, and further details on the algorithm can be found in Supplementary Material SM2.

By combining (3.13) and (3.6) and replacing the refinement $(f_l - f_{l-1})$ with its interpolator $\mathcal{P}_l(x)$, an approximate pointwise error interval of $f_\infty(x)$ can be constructed as

$$\hat{f}_L(x) \pm \left(\frac{|\mathcal{P}_L(x)|}{T^\alpha - 1} + \sum_{l=1}^{L} \sigma_l(x) (\mathbf{z}_l^T \mathbf{\Phi}_l^{-1} \mathbf{z}_l)^{1/2} \right). \tag{3.16}$$

These intervals can be used to quantify emulation uncertainty of the multilevel interpolator.

For the practical choice of the desired prediction accuracy ϵ, one can begin with a large ϵ to conduct the proposed stacking design, and if the prediction performance is not satisfactory (e.g., unsatisfactory predictions when comparing with validation simulations), the prediction accuracy can be subsequently improved by performing a "post" stacking design. More precisely, one can increase the precision by further selecting a smaller ϵ and iterating the above steps of the algorithm starting from the fidelity level L, at which the previous stacking design was terminated. This can be naturally done because the additional design points can be stacked on the previous stacking design.

4. Cost complexity theorem.

With this in hand, we now investigate the computational cost for training data simulation to achieve the desired prediction error of ϵ in the following novel theorem. It should be noted that this theorem does not specify which design is used; it shows the *existence* of multifidelity designs that achieve the asserted computational complexity with desired prediction error, and provides useful insights on conditions under which multifidelity emulation may be most effective. Such existence results are commonly encountered in the deterministic sampling literature (see, e.g., p. 40 of [10] and [29]), and typically serve as a first step for design construction. Our theorem is also similar in spirit to the cost complexity theorems in [15] for MLMC, extended to the multilevel interpolation setting at hand.

Let C_l denote the computational cost required for a single simulation run at fidelity level l, and let $C_{\text{tot}} = \sum_{l=1}^{L} n_l C_l$, which is the total computational budget for training data simulation. Our cost complexity theorem is stated as follows.

Theorem 4.1. *Suppose Ω is bounded and convex, with Φ_l taken as the Matérn kernel (2.3). Further suppose the smoothness parameters $\nu_l = \nu$ for each level l. Assume there exist positive constants $\alpha \geq \beta\nu/d, \beta, c_1, c_2, c_3$, and c_4 such that, for $l = 1, \ldots, L$,*

1. *the* simulation error *$|f_\infty(x) - f_l(x)|$ is bounded as in (3.10); furthermore, $c_1(x) \leq c_1$ for all $x \in \Omega$ and all $l = 1, \ldots, L$;*
2. *the* refinement function *$(f_l - f_{l-1}) \in \mathcal{N}_{\Phi_l}(\Omega)$, and there exists $v_l \in L_2(\Omega)$, such that $f_l(x) - f_{l-1}(x) = \int_\Omega \Phi_l(x,y) v_l(y) \mathrm{d}y$, and $\bar{v} := \sup_{l \in \mathbb{N}^+} \|v_l\|_{L_2(\Omega)} < +\infty$;*
3. *the* kernel length-scale *parameters are bounded as $\|\Theta_l^{-1}\|_2 < c_2$;*
4. *the* designs *$\mathcal{X}_l$ are quasi-uniform, i.e., $h_{\mathcal{X}_l} < c_3 n_l^{-1/d}$;*
5. *the* computational cost *C_l is bounded as $C_l \leq c_4 \xi_l^{-\beta}$.*

Assuming an error tolerance of $0 < \epsilon < e^{-1}$, there then exist choices of L and $n_1, \ldots, n_L$ for which the multilevel interpolator achieves the desired prediction bound

$$|f_\infty(x) - \hat{f}_L(x)| \leq \epsilon, \quad x \in \Omega,$$

with a total computational cost *C_{tot} bounded by*

$$C_{\text{tot}} \leq \begin{cases} c_5 \epsilon^{-\frac{d}{\nu}}, & \frac{\alpha}{\beta} > \frac{2\nu}{d}, \\ c_5 \epsilon^{-\frac{d}{\nu}} |\log \epsilon|^{1+\frac{d}{\nu}}, & \frac{\alpha}{\beta} = \frac{2\nu}{d}, \\ c_5 \epsilon^{-\frac{d}{\nu} - \frac{2\beta\nu - \alpha d}{2\alpha(\nu+d)}}, & \frac{\alpha}{\beta} < \frac{2\nu}{d}, \end{cases} \tag{4.1}$$

where c_5 is a positive constant.

Note that condition 2 implies that the extended function $(f_l - f_{l-1})_e \in H^{2\nu+d}(\mathbb{R}^d)$. This higher-order smoothness requirement allows for improved convergence rates in our specific context; see [48] for further details. The proof of Theorem 4.1 is given in Supplementary Material SM3. The underlying principle of the proof is to require L to be

$$L = \left\lceil \frac{\log(2c_1 \xi_0^\alpha \epsilon^{-1})}{\alpha \log T} \right\rceil$$

to ensure $|f_\infty(x) - f_L(x)| \leq \epsilon/2$, where $\lceil \cdot \rceil$ rounds up to the nearest integer and $c_1 = \sup_{x \in \Omega} c_1(x)$, and then select the optimal $n_1, \ldots, n_L$,

$$n_l \propto \xi_l^{\frac{(\alpha+2\beta)d}{2(\nu+d)}}, \tag{4.2}$$

and a constant of proportionality is chosen so that $|f_L(x) - \hat{f}_L(x)| \leq \epsilon/2$. In practice, the value of L is unknown due to the presence of the unknown constant c_1. This is where our proposed stacking design comes into play in practice, as it can be utilized to determine the value of L effectively.

While this theorem is quite involved, it provides several novel and useful insights on the multifidelity design problem. By (4), for a given fidelity level l, it follows that the computational work $n_l \xi_l^{-\beta}$ satisfies

$$n_l \xi_l^{-\beta} \propto \xi_l^{\frac{\alpha d - 2\beta\nu}{2(\nu+d)}}, \quad l = 1, \ldots, L, \tag{4.3}$$

where ξ_l is again the fidelity parameter for level l. From this, a key factor for determining how much of the total budget C_{tot} to allocate to each fidelity is whether the numerator of the last term $\alpha d - 2\beta\nu > 0$ or, equivalently, the factor $\alpha/\beta > 2\nu/d$. When $\alpha/\beta > 2\nu/d$, one can see from (4.3) that much of the budget C_{tot} is expended on the levels with lower fidelities, i.e., with coarser mesh densities. Conversely, when $\alpha/\beta < 2\nu/d$, much of the budget will be allocated to levels with higher fidelities, i.e., with denser mesh densities.

One can further glean intuition on the terms α/β and $2\nu/d$ in this comparison. Recall that the parameters α, β, ν, and d correspond to the rate parameter for simulator error convergence (see (3.10)), the rate of increase in computational cost C_l as fidelity increases, the smoothness of the refinement function $(f_l - f_{l-1})$, and the number of input parameters, respectively. One can thus interpret the first fraction α/β as the rate of simulator error reduction over the rate of computational cost increase as fidelity increases. Similarly, the second fraction $2\nu/d$ can be interpreted as the rate of convergence for the RKHS interpolator (see, e.g., [52]). Thus, when α/β exceeds this rate of convergence for the RKHS interpolator (due to a combination of (i) and (ii)), the design procedure would shift more computational resources towards lower-fidelity simulation runs, which is quite intuitive.

To further explore this idea, we compare next the cost complexity rate in Theorem 4.1 with the corresponding rate if our emulator were trained using only high-fidelity simulation data. Note that the fidelity level chosen for this latter high-fidelity emulator may be different from the high-fidelity level L for the multifidelity interpolator; to distinguish this, we will use fidelity level H with corresponding fidelity parameter ξ_H. For this high-fidelity emulator, its predictor is given by the RKHS interpolator of $f_H|_{\mathcal{X}_H}$, i.e.,

$$\hat{f}_H(x) = \Phi_H(x, \mathcal{X}_H)\mathbf{\Phi}_H^{-1}\mathbf{y}_H, \tag{4.4}$$

where $\mathbf{y}_H := f_H|_{\mathcal{X}_H}$ is the response vector simulated at fidelity level H. The following corollary shows a similar cost complexity result for this high-fidelity RKHS interpolator:

Corollary 4.2. *Assume that*

1. *there exist some $\xi_H > 0$ and $0 < \epsilon < 1$ for which $(\epsilon/2)^{1+\frac{\alpha d}{2\nu\beta}} \leq c_1\xi_H^\alpha \leq \epsilon/2$, where $c_1 = \sup_{x\in\Omega} c_1(x)$;*
2. *the high-fidelity* response surface *$f_H \in \mathcal{N}_{\Phi_H}(\Omega)$ and the extended function $(f_H)_e \in H^{2\nu+d}(\mathbb{R}^d)$;*
3. *there exists a positive constant c_2 such that the* kernel length-scale *parameter is bounded as $\|\Theta_H^{-1}\|_2 < c_2$.*

There then exists a sample size n_H for which the high-fidelity emulator (4.4) achieves the desired prediction bound

$$|f_\infty(x) - \hat{f}_H(x)| \leq \epsilon, \quad x \in \Omega,$$

with a total computational cost *C_H bounded by*

$$C_H \leq c_h\epsilon^{-\frac{\beta}{\alpha}-\frac{d}{2\nu}}, \tag{4.5}$$

where c_h is a positive constant.

By comparing the above rate with the complexity rate (4.1) for the multilevel interpolator (Theorem 4.1), one can gain illuminating insights into when multifidelity emulation improves upon standard high-fidelity RKHS interpolators. When $\alpha/\beta < 2\nu/d$ (the same ratio compared earlier), the multilevel interpolator rate improves upon the high-fidelity interpolator rate and, conversely, when $\alpha/\beta \geq 2\nu/d$, the high-fidelity rate is quicker than the multilevel rate. Such a condition is intuitive and can be reasoned from the rate parameters α and β. For α, take the limiting setting of $\alpha \to \infty$, such that the simulation error (3.10) decreases *rapidly* to zero as fidelity increases. In this setting, it is intuitive that a high-fidelity interpolator (which relies solely on such high-accuracy runs) would outperform the multilevel interpolator; this is then affirmed by the fact that the condition $\alpha/\beta \geq 2\nu/d$ is satisfied. For β, take the limiting setting of $\beta \to 0$, such that the computational cost C_l grows *slowly* as fidelity increases. In this case, it makes sense that a high-fidelity interpolator (which would not be costly) would outperform the multilevel interpolator given a fixed budget; this is again affirmed by the condition $\alpha/\beta \geq 2\nu/d$. Similar conclusions also hold in reverse: when $\alpha \to 0$ or $\beta \to \infty$, i.e., when the simulation error (3.10) decreases rapidly to zero or the cost C_l grows rapidly as fidelity increases, analogous reasoning can be used to explain why the multilevel interpolator would be more preferable than the high-fidelity interpolator given a fixed cost budget. In this view, Theorem 4.1 provides a novel perspective on when multifidelity modeling improves upon high-fidelity modeling for emulation. We note that the analysis is based on the assumption of a small error ϵ that satisfies $\epsilon < 2c_1\xi_1^\alpha$, which ensures the necessity of at least two fidelity levels for the multifidelity computer experiments in Theorem 4.1.

It should be noted that, in the above analysis, it is assumed that one knows the high-fidelity fidelity parameter ξ_H such that Condition 1 in Corollary 4.2 is satisfied. For practical implementation, such a fidelity parameter is typically not known, and a misspecification of this ξ_H can lead to a worse cost complexity rate than what is guaranteed by Corollary 4.2 for the high-fidelity interpolator. The proposed stacking design gets around this issue of "fidelity misspecification," by employing a sequential sampling approach for determining the number of levels L and corresponding sample sizes $n_1, \ldots, n_L$ to achieve the desired error tolerance.

Finally, we note that the above analysis is based on the established *upper* bound on computational cost C_{tot} in (4.1). This might be made more concrete with a matching *lower* bound on C_{tot}; this would preclude the existence of designs with the same accuracy but lower cost complexity, and provide a tight cost complexity bound for analysis. However, such a lower bound does not appear to be straightforward, and we thus defer this to future work. The employed upper-bound cost analysis has been widely used in the analysis of MLMC methods [15, 16], where useful insights have been gleaned; our analysis above does the same for the related problem of multilevel emulation.

5. Numerical experiments.

We now investigate a suite of numerical experiments to examine the proposed stacking designs, in particular, its predictive performance and its ability to achieve a desired error tolerance. Section 5.1 explores a synthetic example, Section 5.2 investigates an application to Poisson's equation, and section 5.3 considers an application for thermal stress analysis of a jet engine turbine blade. The latter two problems involve PDE systems which are numerically solved via finite element modeling. These experiments are all initialized with an initial design $\mathcal{X}_0$ of size $n_0 = 5d$. All experiments are performed on a MacBook Pro laptop with Apple M1 Max Chip and 32 Gb of RAM.

5.1. Multifidelity Currin function. We first consider the following two-dimensional multifidelity Currin function:

$$f_l(x_1, x_2) = f_\infty(x_1, x_2) + \xi_l^\alpha \exp(-1.4x_1)\cos(3.5\pi x_2). \tag{5.1}$$

Here, $f_\infty(x_1, x_2)$ is the Currin test function [8] which we take to be our limiting highest-fidelity setting:

$$f_\infty(x_1, x_2) = \left[1 - \exp\left(-\frac{1}{2x_2}\right)\right] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}, \quad (x_1, x_2) \in [0,1]^2 = \Omega.$$

The remaining term in (5.1) is the discrepancy term, which converges to zero as fidelity parameter ξ_l increases and makes (5.1) satisfy the inequality of (3.10). In the following, we set $\alpha = 1$ and $\xi_l = \xi_0 T^{-l} = 16 \times 2^{-l}$, and set the computational cost to be $C_l = 4^l$. We assume that α is unknown which needs to be estimated. Figure 3 shows the synthetic functions $f_l(x_1, x_2)$ for fidelity levels $l = 1, 2, 3$ and the limiting function $f_\infty(x_1, x_2)$. With this, we then applied the stacking design algorithm from section 3.3, with the desired prediction accuracy set as $\epsilon = 1$ in the L_2 norm.

The stacking design begins with $L = 1$, which requires $n_1 = 23$ design points on the lowest fidelity simulator f_1 (left panel of Figure 4) to ensure that the estimated emulation error bound (3.8) of $\|f_1 - \hat{f}_1\|_{L_2(\Omega)}$ is smaller than $\epsilon/2 = 0.5$; this is summarized in the $L = 1$ column in Table 1. In the next step with $L = 2$, we then add on an additional 42 design points for the lower-fidelity simulator f_1, yielding a total of $n_1 = 65$ runs on f_1. We then add $n_2 = 25$ design points on a higher-fidelity simulator f_2 (i.e., fidelity level $l = 2$), which is then "stacked" on top of the lower-fidelity design (second panel from the left of Figure 4). With the designs conducted at these two fidelity levels, the estimated emulation error bound $\|f_2 - \hat{f}_2\|_{L_2(\Omega)}$ comes to 0.455, which is again less than $\epsilon/2 = 0.5$. We then repeat this iteratively for increasing fidelity levels $L = 3$ and $L = 4$, after which the stopping rule (3.14) is satisfied and the procedure is terminated. Figure 4 visualizes the experimental designs and corresponding samples sizes at each step.

To evaluate the simulation error bound in the stopping rule (3.14), the rate parameter α needs to be estimated from data; this can be done via (3.15) when $L \geq 3$. The estimates of α

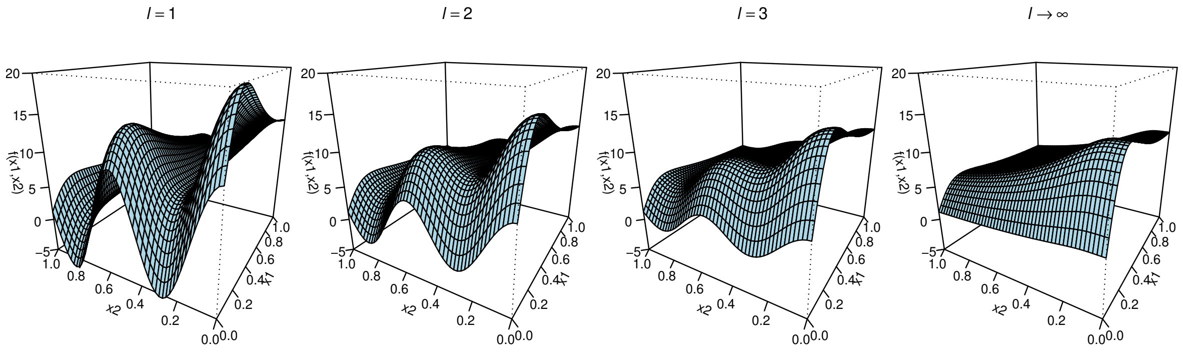

Figure 3. *Visualizing the multifidelity Currin function $f_l(x_1, x_2)$ for $l = 1, 2, 3$ and the limiting (highest-fidelity) Currin function $f_\infty(x_1, x_2)$.*

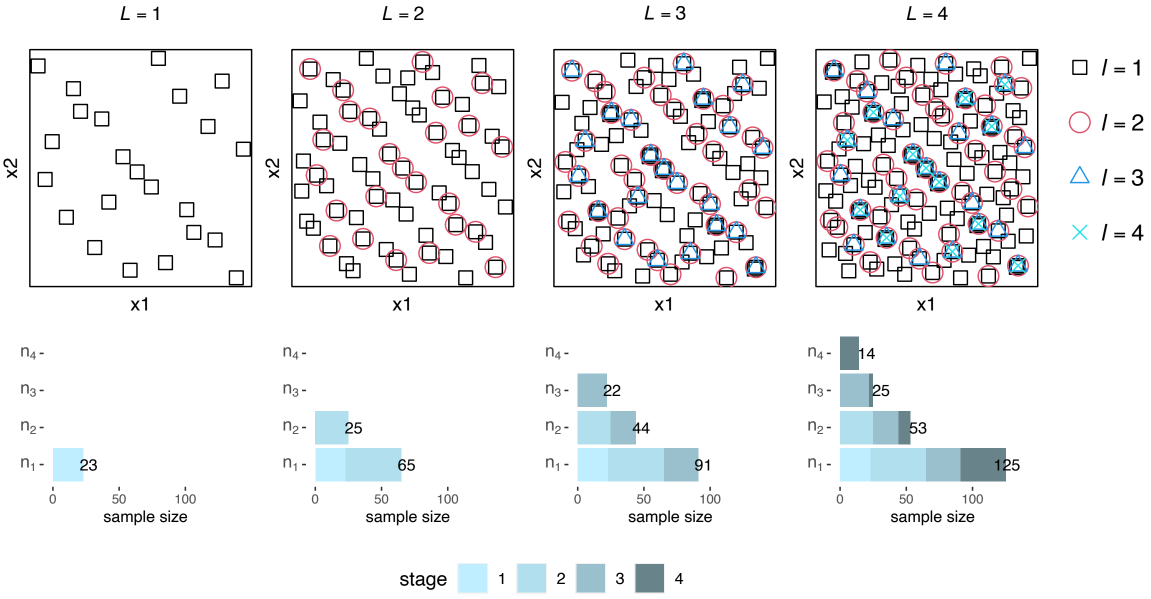

Figure 4. *(Top) The proposed stacking designs at $L = 4$ fidelity levels, over four batch sequential design stages (from left to right). (Bottom) The corresponding sample sizes at the $L = 4$ fidelity levels, over four batch sequential design stages (from left to right).*

Table 1

The estimated simulation and emulation error bounds (see (3.14) and (3.8), respectively) at each design stage for the multifidelity Currin experiment, with estimated rate parameter $\hat{\alpha}$ at stages $L = 3$ and $L = 4$. Bolded numbers indicate the error is less than $\epsilon/2$, where $\epsilon = 1$ is the desired error tolerance.

	$L = 1$	$L = 2$	$L = 3$	$L = 4$
Fidelity parameter	$\xi_1 = 8$	$\xi_2 = 4$	$\xi_3 = 2$	$\xi_4 = 1$
Cost per run	$C_1 = 4$	$C_2 = 16$	$C_3 = 64$	$C_4 = 256$
Bound of $\|f_\infty - f_L\|_{L_2(\Omega)}$	NA	NA	0.798 $(\hat{\alpha} = 1)$	**0.448** $(\hat{\alpha} = 1)$
Bound of $\|f_L - \hat{f}_L\|_{L_2(\Omega)}$	**0.483**	**0.455**	**0.324**	**0.474**

(reported in Table 1) are precisely equal to the true parameter $\alpha = 1$. With the estimate of α at $L = 3$, the simulation error bound (3.14) gives 0.798, which is greater than $\epsilon/2$ and thus the batch sequential design continues. With the estimate of α at $L = 4$, the simulation error estimate is less than $\epsilon/2 = 0.5$, thus the stopping rule is satisfied and the design procedure stops. Table 1 shows the estimated upper bounds for simulation and emulation errors, both of which need to be smaller than $\epsilon/2 = 0.5$ for the procedure to stop. The L_2-error of the final multilevel interpolator (estimated via Monte Carlo integration) is $\|f_\infty - \hat{f}_4\|_{L_2(\Omega)} = 0.53$, which is indeed smaller than the desired prediction accuracy of $\epsilon = 1$. This shows that the proposed stacking designs, by increasing fidelity levels and stacking design points in a sequential fashion, can indeed satisfy the desired error bound.

To benchmark against the state-of-the-art, we further implemented the sequential design strategy in [25] using the above setup, with code provided in the supplementary materials of

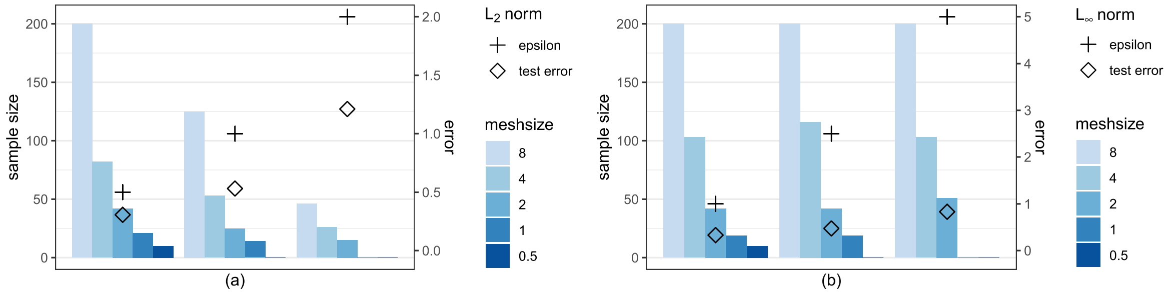

Figure 5. *Visualizing the allocated sample sizes from stacking designs and the corresponding L_2 (left) and L_∞ (right) test errors (marked $\diamond$) at various error tolerances (marked $+$) for the multifidelity Currin experiment.*

their paper. This comparison serves *solely* as a benchmark for gauging predictive accuracy of our approach; the above existing sequential design does not target a desired predictive accuracy $\epsilon > 0$, which is the key focus of our method. As the provided code was specifically designed for two fidelity levels (and is difficult to generalize for more levels), we uniformly sampled two values of ξ_l (from 1 to 8) as the two fidelity levels for simulations, then performed the sequential design using the same total computational cost as our method (which is 6532) as a stopping criterion. This experiment was replicated 100 times. The average emulation L_2-error using the above existing sequential designs is 1.59 (with a standard deviation of 0.73), which is higher than the error using the proposed stacking designs (0.53); thus, our approach appears to offer better (or at least comparable) predictions to this state-of-the-art approach.

Finally, we explore the performance of stacking designs for different choices of error tolerance ϵ, using both the L_2 and L_∞ norms. Figure 5 visualizes the sample sizes (at each fidelity level) and the corresponding errors of the final multilevel interpolator. We see that, for different ϵ and different norms, the proposed stacking designs can consistently yield prediction errors which are smaller than the desired error tolerance ϵ, which is as desired.

5.2. Poisson's equation. Next, we explore the performance of stacking designs for emulating an elliptical PDE system. The system of interest is modeled using Poisson's equation on a square membrane [12], which has broad applicability in electrostatics and fluid mechanics. This can be represented by the PDE

$$\Delta u = (x^2 - 2\pi^2)e^{xz_1}\sin(\pi z_1)\sin(\pi z_2) + 2x\pi e^{xz_1}\cos(\pi z_1)\sin(\pi z_2), \quad (z_1, z_2) \in D, \tag{5.2}$$

where $u(z_1, z_2)$ is the solution of interest, $\Delta = \partial^2/\partial z_1^2 + \partial^2/\partial z_2^2$ is the Laplace operator, $D \in [0,1] \times [0,1]$, and $x \in \Omega = [-1,1]$. One then imposes the Dirichlet boundary condition $u = 0$ on the boundary ∂D. Following [49], FEM is used to solve this system numerically. In our implementation, we make use of the partial differential equation toolbox of [31] to create the geometry and mesh. In the toolbox, one can specify the mesh size for the numerical solver; Figure 6 visualizes the numerical solutions of (5.2) with three different choices of mesh sizes.

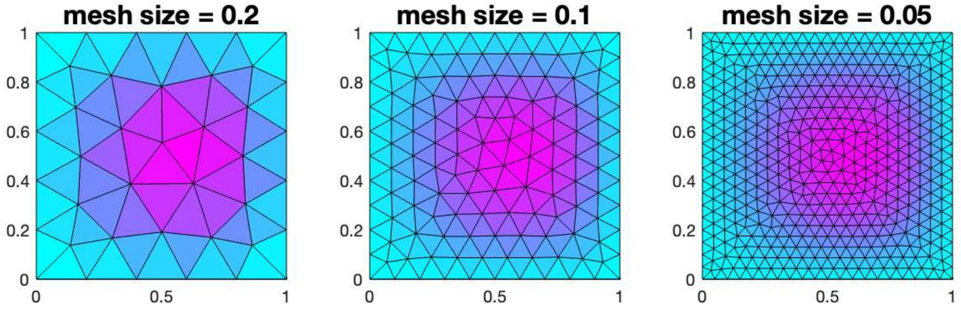

Figure 6. *Visualizing the FEM solutions of Poisson's equation* (5.2) *with three different mesh sizes.*

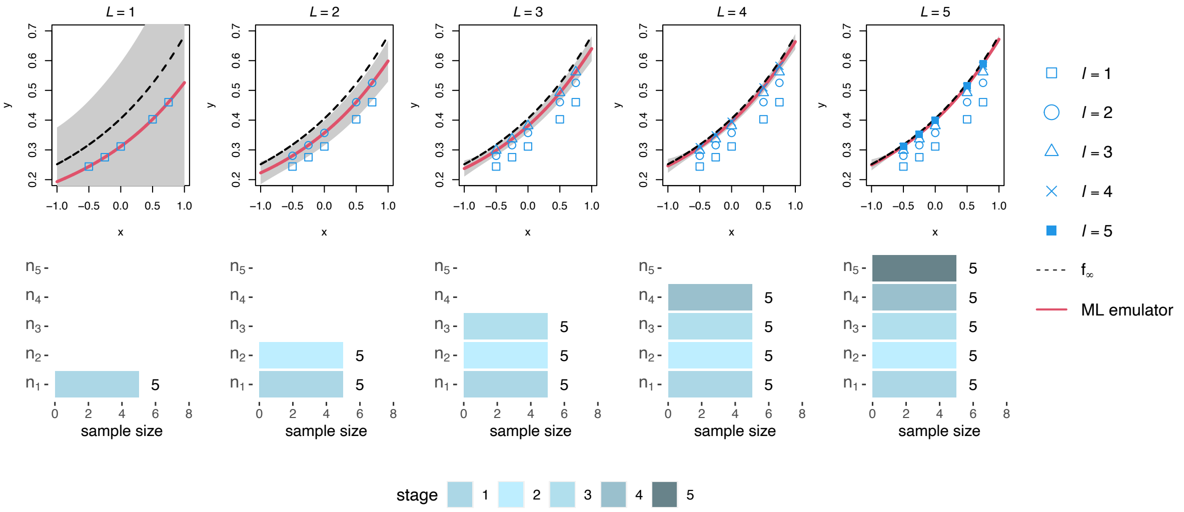

Figure 7. *Visualizing the stacking design procedure for the Poisson's equation experiment. (Top) The stacking design points (dots) and the fitted multilevel interpolator with pointwise error intervals obtained via* (3.16) *(gray shaded region) over five batch sequential design stages (from left to right). (Bottom) The corresponding sample sizes at the $L = 5$ fidelity levels, over five batch sequential design stages (from left to right).*

Here, the response of interest is taken as the integral of the solution $f_\infty(x) = \int u(z_1, z_2) \mathrm{d}z_1 \mathrm{d}z_2$. It can be shown that this solution has the analytical form [49]

$$f_\infty(x) = \frac{2(e^x + 1)}{x^2 + \pi^2}.$$

Of course, in practical problems, one does not typically have such closed-form solutions; this framework was chosen to allow for easy validation of the fitted emulator model. For the stacking designs, we again define a geometric sequence of fidelity parameters (here, mesh sizes) $\xi_l = \xi_0 T^{-l} = 0.4 \times 2^{-l}$, $l = 1, 2, \ldots$. We then set a desired prediction accuracy of $\epsilon = 0.05$ for the L_∞-norm.

Figure 7 shows the fitted multilevel interpolator and pointwise error intervals (upper panels) obtained via (3.16), with its corresponding sample sizes at each fidelity level (bottom

Table 2

The estimated simulation and emulation error bounds (see (3.14) and (3.8), respectively) at each design stage for the Poisson's equation experiment, with estimated rate parameter $\hat{\alpha}$ at stages $L = 3, 4$, and 5. Bolded numbers indicate the error is less than $\epsilon/2$, where $\epsilon = 0.05$ is the desired error tolerance.

	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
Mesh size	$\xi_1 = 0.2$	$\xi_2 = 0.1$	$\xi_3 = 0.05$	$\xi_4 = 0.025$	$\xi_5 = 0.0125$
Cost per run (sec)	$C_1 = 0.18$	$C_2 = 0.19$	$C_3 = 0.23$	$C_4 = 0.27$	$C_5 = 0.55$
Bound of $\Vert f_\infty - f_L\Vert_{L_\infty(\Omega)}$	NA	NA	0.047 ($\hat{\alpha} = 0.92$)	0.029 ($\hat{\alpha} = 0.84$)	**0.007** ($\hat{\alpha} = 1.07$)
Bound of $\Vert f_L - \hat{f}_L\Vert_{L_\infty(\Omega)}$	**0.008**	**0.011**	**0.014**	**0.015**	**0.015**

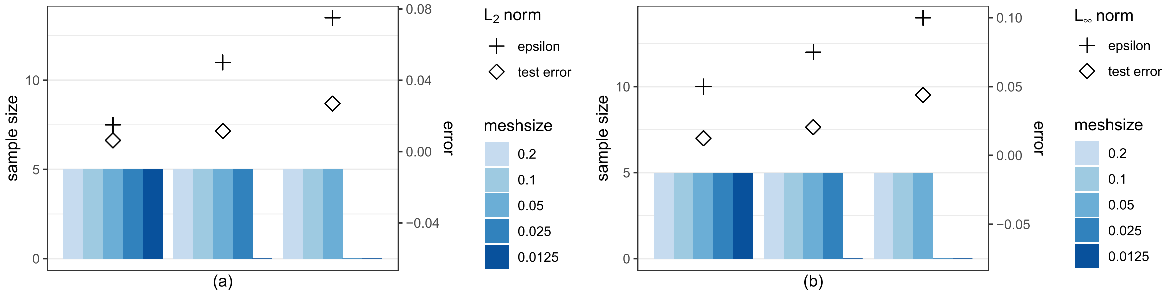

Figure 8. *Visualizing the allocated sample sizes from stacking designs and the corresponding L_2 (left) and L_∞ (right) test errors (marked ◇) at various error tolerances (marked +) for the Poisson's equation experiment.*

panels) at each step of the stacking design process. Table 2 shows the estimated emulation and simulation error bounds. In the first step with $L = 1$, the lowest-fidelity simulation (with mesh size 0.2) is run with a design of size $n_1 = 5$. From the left panels in Figure 7, the resulting fitted multilevel interpolator $\hat{f}_1$ appears to be biased with very wide error bounds. In the third step ($L = 3$), additional design points are stacked on fidelity levels 2 and 3 ($n_1 = 5, n_2 = 5, n_3 = 5$), and the simulation error rate parameter α is then estimated via (3.15). With this estimated parameter, the estimated simulation error bound (3.14) evaluates to 0.047 (see Table 2), which is greater than $\epsilon/2 = 0.025$. The process thus continues until the fifth step ($L = 5$), in which both the simulation and emulation error bounds become smaller than $\epsilon/2$. From the right panels in Figure 7, the resulting fitted emulator $\hat{f}_5$ appears to yield an accurate prediction with narrow error bounds. The L_∞-error of this final multilevel interpolator (estimated via the grid search optimization) is $\Vert f_\infty - \hat{f}_5\Vert_{L_\infty(\Omega)} = 0.013$, which is smaller than the desired prediction accuracy $\epsilon = 0.05$. This again shows that the proposed stacking designs, by iteratively increasing fidelity levels and stacking design runs, can yield the desired error tolerance. We again compare with the sequential design in [25]. The experimental setup is similar to that described in section 5.1. The results indicate that, at the same computational cost, the L_∞-error of the emulator using the sequential designs is 0.058 (with standard deviation of 0.035), which is higher than ours (which is 0.013).

As before, we further explore the stacking designs for this problem with different error tolerances ϵ using both L_2 and L_∞ norms. Figure 8 shows the sample sizes (at each fidelity level) and the corresponding errors of the final multilevel interpolator. We see again that the proposed designs can indeed consistently satisfy the desired error tolerance ϵ. It is worth noting that, in this problem, a majority of the computational budget is expended on higher-fidelity runs (i.e., with denser mesh densities), since the sample sizes are equally allocated over each fidelity level (see Figure 8). This is not too surprising: the fitted function is quite smooth ($\hat{\nu} = 3.5$), and with estimated rate parameters $\hat{\alpha} \approx 1$ and $\hat{\beta} \approx 0.37$, the condition $\alpha/\beta < 2\nu/d$ can be shown to be satisfied (see earlier discussion of this condition in section 4 and the cost complexity theorem).

5.3. Thermal stress analysis of jet engine turbine blade. Finally, we investigate the performance of the proposed designs on a thermal stress analysis application for a jet turbine engine blade in steady-state operating condition. Here, the turbine is a component of the jet engine, composed of a radial array of blades typically made from nickel alloys that resist extremely high temperatures. To avoid mechanical failure and friction between the tip of the blade and the turbine casing, it is crucial that the blade design can withstand stress and deformations. See [53] and [7] for more details. We thus wish to study here the effect of thermal stress and pressure of the surrounding gases on turbine blades. As before, this problem can be analyzed as a static structural model, which can be numerically solved using finite element modeling. The $d = 2$ input variables are the pressure load on the pressure (x_1) and suction (x_2) sides of the blade, both of which range from 0.25 to 0.75 MPa, i.e., $x_1, x_2 \in \Omega = [0.25, 0.75]^2$. The response of interest is taken as the integral of the solution over the thermal stress profile. FEM simulations are performed using thePDE toolbox in MATLAB [31]. Figure 9 visualizes the blade structure and the simulated thermal stress profiles at three choices of mesh sizes.

For stacking designs, we make use of a geometric sequence of fidelity parameters (here, mesh sizes) $\xi_l = \xi_0 T^{-l} = 0.1 \times 2^{-l}$, $l = 1, 2, \ldots$. The desired prediction accuracy is then set to $\epsilon = 5$ in the L_2-norm. Figure 1 shows the proposed stacking designs over each iteration, Table 3 summarizes the corresponding emulation and simulation error bounds. Here, we see that the multilevel interpolator requires $L = 4$ iterations (resulting in $L = 4$ fidelity levels for the final emulator) to achieve the desired prediction accuracy. Unlike in section 5.2, the

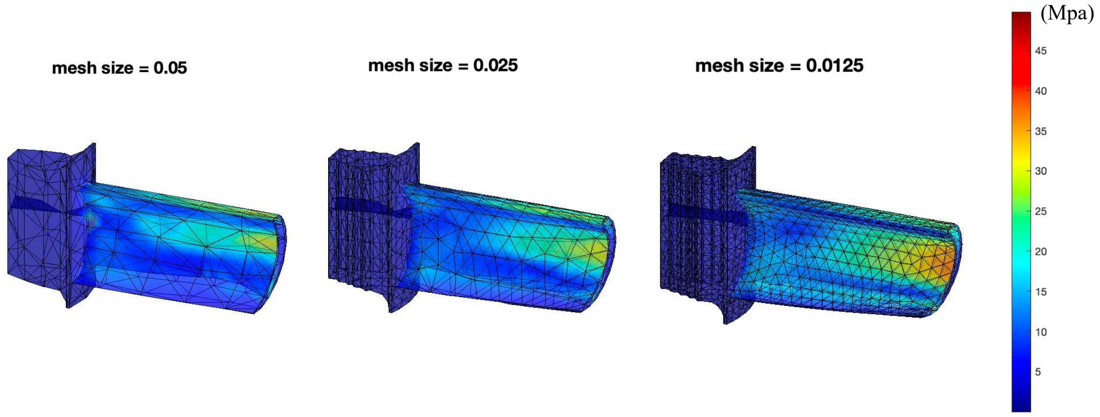

Figure 9. *Visualizing the FEM solutions for three choices of mesh sizes in the turbine blade application.*

Table 3

The estimated simulation and emulation error bounds (see (3.14) and (3.8), respectively) at each design stage for the turbine blade application, with estimated rate parameter $\hat{\alpha}$ at stages $L = 3$ and $L = 4$. Bolded numbers indicate the error is less than $\epsilon/2$, where $\epsilon = 5$ is the desired error tolerance.

	$L=1$	$L=2$	$L=3$	$L=4$
Mesh size	$\xi_1 = 0.05$	$\xi_2 = 0.025$	$\xi_3 = 0.0125$	$\xi_4 = 0.00625$
Cost per run (sec.)	$C_1 = 0.75$	$C_2 = 1.07$	$C_3 = 2.13$	$C_4 = 11.51$
Bound of $\Vert f_\infty - f_L \Vert_{L_2(\Omega)}$	NA	NA	2.969 $(\hat{\alpha} = 0.81)$	**0.956** $(\hat{\alpha} = 1.09)$
Bound of $\Vert f_L - \hat{f}_L \Vert_{L_2(\Omega)}$	**2.324**	**2.408**	**2.481**	**2.491**

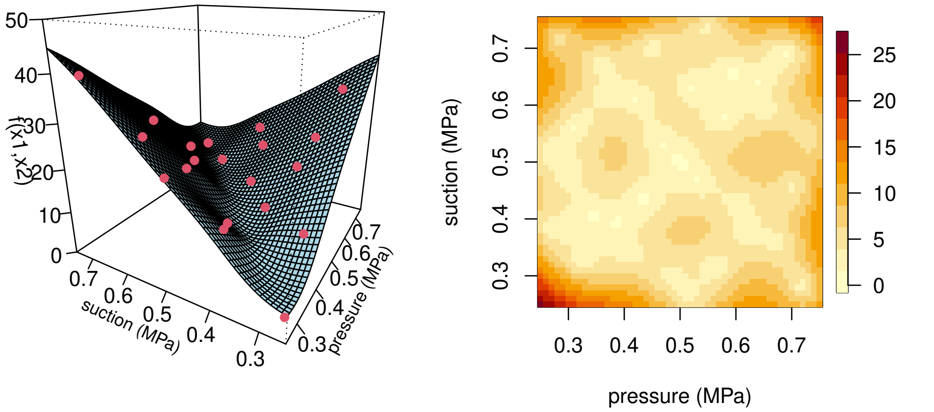

Figure 10. *(Left) Visualizing the fitted multilevel interpolator with the test design points in red. (Right) Visualizing the pointwise error bounds (3.16) over the design space.*

true function $f_\infty(x)$ cannot be expressed in closed form; we thus perform validation runs at 20 uniformly sampled input settings with a small mesh size of $\xi_5 = 3.125 \times 10^{-3}$. Figure 10 visualizes the final multilevel interpolator $\hat{f}_L(x_1, x_2)$ as in (2.2) with the out-of-sample test points (red points), as well as the pointwise error bounds (3.16) over the input space. We see that the predicted response surface quite closely mimics the test data, which is as desired. This is confirmed by the empirical L_2-norm of the prediction error on the test data (1.60), which is smaller than the desired error tolerance of $\epsilon = 5$. This again shows that the proposed stacking designs can indeed achieve the desired prediction accuracy via iterative multifidelity modeling. Similarly to previous subsections, we compare with the sequential design in [25] with a similar experimental setup. The results indicate that, at the same computational cost, the L_2-error of the emulator using the sequential designs is 2.12 (with standard deviation of 0.83), which is higher than ours (which is 1.60).

6. Concluding remarks.

In this work, we proposed a novel stacking design approach for multifidelity modeling, which provides a sequential approach for designing multifidelity runs to achieve a desired prediction error $\epsilon > 0$. This addresses a key limitation of existing design methods, and allows for confident and cost-efficient predictive computing for a broad range of scientific and engineering problems. We then proved a cost complexity theorem which, under the employed multilevel RKHS interpolator, establishes a bound on computational cost (of the training data simulation) needed to ensure a desired prediction error tolerance of ϵ. A corollary then provides new insight on when the presented multilevel interpolator yields improved predictive performance over a single-fidelity RKHS interpolator. A suite of numerical experiments, including an application to jet turbine blade analysis, shows that the proposed method can efficiently and accurately emulate multifidelity computer experiments with some notion of confidence.

It is worth noting that alternative models, such as the one proposed by [49], offer interesting possibilities for addressing the challenges of multifidelity emulation. This model, which incorporates the rate at which the error with respect to the ideal/exact solution decreases, presents a different approach that could potentially enhance the efficiency of the emulation process. Future research can explore the utilization of this model to derive a similar error bound and develop a corresponding design methodology. In addition, we have identified recent studies, such as [51, 47, 54], which consider the impact of parameter misspecifications in the GPs and provide important extensions of stacking designs from a Bayesian perspective. In particular, these bounds can be used to investigate the emulation error in a GP assumption and determine sample sizes under the constraints of a given computational budget (instead of a given target error tolerance), as demonstrated in [11]. Future research directions could involve exploring the optimal number of fidelity levels L within a given computational budget. Finally, the development of potentially matching lower bounds for our cost-complexity theory would be useful in strengthening the insightful conclusions in section 4; we will investigate this as future work.

Supplemental materials. Additional supporting materials can be found in the supplemental materials, including the detailed proofs of Proposition 3.1, Theorem 4.1, and Corollary 4.2, the detailed algorithm for section 3.3, and the `R` code for reproducing the results in section 5.

Acknowledgments. The authors gratefully acknowledge the conscientious efforts of the associate editor and two anonymous reviewers whose comments greatly strengthen this paper.

REFERENCES

[1] R. Ahlswede and I. Wegener, *Search Problems*, Wiley, Chichester, England, 1987.

[2] J. Bect, S. Zio, G. Perrin, C. Cannamela, and E. Vazquez, *On the quantification of discretization uncertainty: Comparison of two paradigms*, in 14th World Congress in Computational Mechanics and ECCOMAS Congress 2020 (WCCM-ECCOMAS), 2021.

[3] A. Breger, M. Ehler, and M. Gräf, *Points on manifolds with asymptotically optimal covering radius*, J. Complexity, 48 (2018), pp. 1–14.

[4] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods,* 3rd ed., Springer, New York, 2007.

[5] R. E. Caflisch, *Monte Carlo and quasi-Monte Carlo methods*, Acta Numer., 7 (1998), pp. 1–49.

[6] S. Cao, et al., *Determining the jet transport coefficient $\hat{q}$ from inclusive hadron suppression measurements using Bayesian parameter estimation*, Phys. Rev. C (3), 104 (2021), 024905.

[7] T. J. Carter, *Common failures in gas turbine blades*, Eng. Fail. Anal., 12 (2005), pp. 237–247.

[8] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, *A Bayesian Approach to the Design and Analysis of Computer Experiments*, Technical report, ORNL-6498, Oak Ridge National Lab., TN, 1988.

[9] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, *Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments*, J. Amer. Statist. Assoc., 86 (1991), pp. 953–963.

[10] J. Dick, F. Y. Kuo, and I. H. Sloan, *High-dimensional integration: The quasi-Monte Carlo way*, Acta Numer., 22 (2013), pp. 133–288.

[11] A. Ehara and S. Guillas, *An adaptive strategy for sequential designs of multilevel computer experiments*, Int. J. Uncertain. Quantif., 13 (2023), pp. 61–98.

[12] L. C. Evans, *Partial Differential Equations,* 2nd ed. Grad. Stud. Math. 19, American Mathematical Society, Providence, RI, 2010.

[13] D. Everett, et al., *Multisystem Bayesian constraints on the transport coefficients of QCD matter*, Phys. Rev. C (3), 103 (2021), 054904.

[14] K.-T. Fang and Y. Wang, *Number-Theoretic Methods in Statistics*, Monogr. Statist. Appl. Probab. 51, Chapman & Hall, London, 1993.

[15] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617.

[16] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328.

[17] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*, Chapman and Hall/CRC, Boca Raton, FL, 2020.

[18] B. Haaland and P. Z. G. Qian, *Accurate emulators for large-scale computer experiments*, Ann. Statist., 39 (2011), pp. 2974–3002.

[19] W. H. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-dependent Advection-Diffusion-Reaction Equations*, Springer Ser. Comput. Math. 33, Springer, Berlin, 2003.

[20] Y. Ji, S. Mak, D. Soeder, J.-F. Paquet, and S. A. Bass, *A graphical multi-fidelity Gaussian process model, with application to emulation of expensive computer simulations*, Technometrics, to appear, 2022.

[21] Y. Ji, H. S. Yuchi, D. Soeder, J.-F. Paquet, S. A. Bass, V. R. Joseph, C. Wu, and S. Mak, *Multi-stage multi-fidelity Gaussian process modeling, with application to heavy-ion collisions*, SIAM/ASA J. Uncertain. Quantif., (2014), in press.

[22] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, *Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences*, preprint, arXiv:1807.02582, 2018.

[23] M. C. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, Biometrika, 87 (2000), pp. 1–13.

[24] L. Le Gratiet, *Bayesian analysis of hierarchical multifidelity codes*, SIAM/ASA J. Uncertain. Quantif., 1 (2013), pp. 244–269.

[25] L. Le Gratiet and C. Cannamela, *Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes*, Technometrics, 57 (2015), pp. 418–427.

[26] L. Le Gratiet and J. Garnier, *Recursive co-kriging model for design of computer experiments with multiple levels of fidelity*, Int. J. Uncertain. Quantif., 4 (2014), pp. 365–386.

[27] M. Lukić and J. Beder, *Stochastic processes with sample paths in reproducing kernel Hilbert spaces*, Trans. Amer. Math. Soc., 353 (2001), pp. 3945–3969.

[28] S. Mak and V. R. Joseph, *Minimax and minimax projection designs using clustering*, J. Comput. Graph. Statist., 27 (2018), pp. 166–178.

[29] S. Mak and V. R. Joseph, *Support points*, Ann. Statist., 46 (2018), pp. 2562–2592.

[30] S. Mak, C.-L. Sung, X. Wang, S.-T. Yeh, Y.-H. Chang, V. R. Joseph, V. Yang, and C. F. J. Wu, *An efficient surrogate model for emulation and physics extraction of large eddy simulations*, J. Amer. Statist. Assoc., 113 (2018), pp. 1443–1456.

[31] Mathworks, *MATLAB version* 9.11.0.1769968 (R2021b), Mathworks, Natick, MA, 2021.

[32] S. Müller, *Komplexität und Stabilität von kernbasierten Rekonstruktionsmethoden*, Ph.D. thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, Göttingen, Germany, 2009.

[33] S. Myren and E. Lawrence, *A comparison of Gaussian processes and neural networks for computer model emulation and calibration*, Stat. Anal. Data Min., 14 (2021), pp. 606–623.

[34] W. L. Oberkampf and C. J. Roy, *Verification and Validation in Scientific Computing*, Cambridge University Press, New York, 2010.

[35] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis, *Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling*, Proc. A, 473 (2017), 20160751.

[36] P. Z. G. Qian and C. F. J. Wu, *Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments*, Technometrics, 50 (2008), pp. 192–204.

[37] L. F. Richardson, *The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, Philos. Trans. Roy. Soc. A, 210 (1911), pp. 307–357.

[38] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–423.

[39] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments,* 2nd ed., Springer, New York, 2018.

[40] M. Scheuerer, R. Schaback, and M. Schlather, *Interpolation of spatial data–a stochastic or a deterministic problem?*, European J. Appl. Math., 24 (2013), pp. 601–629.

[41] I. M. Sobol', *On the distribution of points in a cube and the approximate evaluation of integrals*, Zh. Vychisl. Mat. Mat. Fiz., 7 (1967), pp. 784–802.

[42] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York, 1999.

[43] R. Stroh, J. Bect, S. Demeyer, N. Fischer, D. Marquis, and E. Vazquez, *Sequential design of multi-fidelity computer experiments: Maximizing the rate of stepwise uncertainty reduction*, Technometrics, 64 (2022), pp. 199–209.

[44] A. L. Teckentrup, *Convergence of Gaussian process regression with estimated hyper-parameters and applications in Bayesian inverse problems*, SIAM/ASA J. Uncertain. Quantif., 8 (2020), pp. 1310–1337.

[45] J. A. Templeton, et al., *Calibration and Forward Uncertainty Propagation for Large-eddy Simulations of Engineering Flows*, Technical report, SAND2015-7938, Sandia National Laboratory, Livermore, CA, 2015.

[46] R. K. Tripathy and I. Bilionis, *Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification*, J. Comput. Phys., 375 (2018), pp. 565–588.

[47] R. Tuo and W. Wang, *Kriging prediction with isotropic Matérn correlations: Robustness and experimental designs*, J. Mach. Learn. Res., 21 (2020), pp. 7604–7641.

[48] R. Tuo, Y. Wang, and C. F. J. Wu, *On the improved rates of convergence for Matèrn-type kernel ridge regression with application to calibration of computer models*, SIAM/ASA J. Uncertain. Quantif., 8 (2020), pp. 1522–1547.

[49] R. Tuo, C. F. J. Wu, and D. Yu, *Surrogate modeling of computer experiments with different mesh densities*, Technometrics, 56 (2014), pp. 372–380.

[50] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 59, SIAM, Philadelphia, 1990.

[51] W. Wang, R. Tuo, and C. F. J. Wu, *On prediction properties of kriging: Uniform error bounds and robustness*, J. Amer. Statist. Assoc., 115 (2020), pp. 920–930.

[52] H. Wendland, *Scattered Data Approximation*, Cambridge Monogr. Appl. Comput. Math. 17, Cambridge University Press, Cambridge, 2004.

[53] L. M. Wright and J.-C. Han, *Enhanced internal cooling of turbine blades and vanes*, Gas Turbine Handb., 4 (2006), pp. 1–5.

[54] G. Wynne, F.-X. Briol, and M. Girolami, *Convergence guarantees for Gaussian process means with misspecified likelihoods and smoothness*, J. Mach. Learn. Res., 22 (2021), pp. 1–40.

[55] D. Xiu, *Numerical methods for stochastic computations*, in Numerical Methods for Stochastic Computations, Princeton University Press, Princeton, NJ, 2010.

[56] S. Yakowitz, P. L'ecuyer, and F. Vazquez-Abad, *Global stochastic optimization with low-dispersion point sets*, Oper. Res., 48 (2000), pp. 939–950.