Greedy nonlinear autoregression for multifidelity computer models at different scales

W. Xing [b], M. Razi [b], R.M. Kirby [b], K. Sun [a], A.A. Shah [a,*]

[a] *School of Energy and Power Engineering, Chongqing University, 174 Shazhengjie, Shapingba, Chongqing 400044, China*
[b] *Scientific Computing and Imaging Institute, University of Utah, 72 S Central Campus Drive, Room 3750 Salt Lake City, UT 84112, United States*

HIGHLIGHTS

- A general Bayesian framework for multi-fidelity simulation using kernel substitution.
- Combines the advantages of NAR and the stochastic collocation.
- predictions of high-fidelity outputs do not require extra low-fidelity simulations.
- A noise-free assumption extends the method to high-dimensional data.
- Sequential learning avoids the need of a special experimental design.

GRAPHICAL ABSTRACT
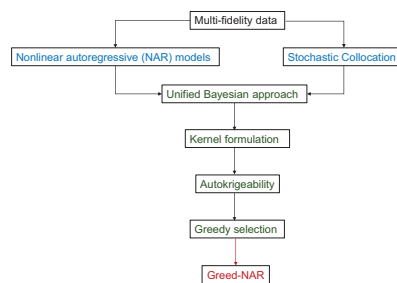
ABSTRACT

Although the popular multi-fidelity surrogate models, stochastic collocation and nonlinear autoregression have been applied successfully to multiple benchmark problems in different areas of science and engineering, they have certain limitations. We propose a uniform Bayesian framework that connects these two methods allowing us to combine the strengths of both. To this end, we introduce Greedy-NAR, a nonlinear Bayesian autoregressive model that can handle complex between-fidelity correlations and involves a sequential construction that allows for significant improvements in performance given a limited computational budget. The proposed enhanced nonlinear autoregressive method is applied to three benchmark problems that are typical of energy applications, namely molecular dynamics and computational fluid dynamics. The results indicate an increase in both prediction stability and accuracy when compared to those of the standard multi-fidelity autoregression implementations. The results also reveal the advantages over the stochastic collocation approach in terms of accuracy and computational cost. Generally speaking, the proposed enhancement provides a straightforward and easily implemented approach for boosting the accuracy and efficiency of concatenated structure multi-fidelity simulation methods, e.g., the nonlinear autoregressive model, with a negligible additional computational cost.

1. Introduction

Computational models are well established tools for the design, development and study of energy technologies, including their constituent components and materials [29,32]. They range from systems-level models for control and automation to *ab-initio* models for materials screening. Even for a given task at a particular spatio-temporal scale, there is typically a large set of approaches available, differing in, primarily, the level of detail included in the underlying model, the particular numeri-

cal techniques employed, and the selection of numerical parameters that control the truncation errors. This results in different levels of computational complexity and attendant time cost, which are strongly correlated with the accuracy of the solutions obtained.

In a given modelling task, we may classify competing computer models, depending on their accuracy and associated complexity, as either *low-fidelity* (lower accuracy, lower complexity) or *high-fidelity*. We may even introduce three or more fidelity levels and classify the various computer models available accordingly. Selecting an approach based on its

* Corresponding author.
  E-mail address: akeelshah@cqu.edu.cn (A.A. Shah)..

fidelity will inevitably involve a trade-off. While high-fidelity models are more accurate, they are (usually) computationally expensive, difficult to implement and difficult to understand for practitioners. Low-fidelity models, on the other hand, can provide rapid solutions and are relatively straightforward to implement, the price for which is a possibly unacceptable level of accuracy.

Modelling tasks such as computer-based optimization of an energy technology or the screening of energy materials require extensive exploration of a design space. In such cases, the computer models are frequently replaced with computationally cheaper approximations, termed *surrogate models*, constructed *via* machine learning methods, projection schemes to lower dimensionality or by combining models of different fidelity (multi-fidelity approaches). Multi-fidelity approaches can also be combined with machine-learning and reduced-order modelling approaches.

Classical machine-learning approaches include Gaussian Process (GP) models, artificial neural networks, support vector machines and polynomial response surface models [15,26], while reduced-order models are typically based on proper orthogonal decomposition or Krylov subspaces [10,17,28]. GP models can be extended to multi-output problems of the type considered here in a number of ways, including treating the output index as an additional input parameter [16], multi-dimensional GP priors with a linearly separable covariance [4], and dimensionality reduction with separate regression on vector components in the resulting low-dimensional linear subspace of physical or feature space [13,34]. Both machine-learning and reduced-order model approaches typically require large training data sets based on high-fidelity models.

In most practical scenarios, however, computational resources are limited, and often insufficient for the acquisition of a large volume of high-fidelity data with acceptable accuracy (often in practice there is even a limited availability of low-fidelity data). Hence, constructing surrogate models that rely only on high-fidelity models may not be desirable, while low-fidelity models lack sufficient accuracy.

One effective way of resolving this dilemma is to use a combination of both low- and high-fidelity data. Although low-fidelity samples are noisy and biased, they normally show a strong correlation with the high-fidelity samples. It can be possible to harness this correlation to avoid full reliance on high-fidelity data. A prominent example of this type of surrogate model combines the stochastic collocation methodology with a multi-fidelity approach [20,35]. Examples of its effectiveness include: frequency-modulated trigonometric functions [20], heat driven cavity flows [11], acoustic horn problems [35], molecular dynamics simulation [24], parametric studies of NACA airfoils [30], discrete-space evolution probability simulations [23], and irradiated particle-laden turbulence [14].

This surrogate modeling approach is also designed to work with a limited number of samples and provide an optimal sampling strategy for the high-fidelity simulations. The authors in [20] use a greedy procedure to select amongst low-fidelity data in order to identify input samples at which to conduct a low number of high-fidelity simulations. They then use the low-fidelity data to find coefficients for an interpolation at out-of-sample points using the high-fidelity results. The reasoning behind the use of low-fidelity data in the diagnostic context of importance sampling points stems from the existence of an expression for the upper bound of the multi-fidelity emulator error, which is a function of the low-fidelity stochastic collocation model error [12]. The limitation of this approach is that it is based on an assumption of the low- and high-fidelity data sharing a similar correlation structure, which limits its application to many complex problems. The other limitation is that it requires an execution of the low-fidelity model when making predictions of high-fidelity outputs. This severely restricts its application when the low-fidelity simulations are also expensive to run.

Autoregressive models in statistics have also been used to construct multi-fidelity emulators. Kennedy and O'Hagan in their seminal work of [15] proposed this approach in its original form with the assumption of a linear relationship between different fidelity levels. The low-fidelity correlations are captured based on low-fidelity observations and transferred over to enhance the high-fidelity model. This method was improved upon in [8] using a deterministic parametric form of the mapping (from low to high fidelity) and an efficient numerical scheme to reduce the computational cost. Despite its success in several experiments, this parametric approach requires expert knowledge for model selection as well as a large dataset for model training.

To overcome the limitations of the linear assumption, a nonlinear autoregressive model (NAR) was introduced in [21] by replacing the linear transformation with a GP model. NAR has been successfully applied to a number of problems. In these enhanced multi-fidelity autoregressive approaches, in order to fully capture and propagate the uncertainty through all fidelity levels, a chain of GP models is jointly trained in a deep GP framework [6]. In recent work, the NAR model was generalised for efficient emulation of high-dimensional output (order 1 million) simulation problems [33].

Despite the success of NAR and its variants, they lack a principled way to sample parsimoniously from expensive high-fidelity simulations, leading to a potential waste of computational resources on the generation of very similar high-fidelity results. Although experimental design techniques, e.g., Latin hypercube [2] and Sobol sequences [31], can be used to improved sampling efficiency, they do not incorporate any particular knowledge of a specific problem and inevitably, therefore, lead to inefficiency.

In this paper, we first derive a general Bayesian framework for multi-fidelity simulation based on the kernel extension of a general linear model. This framework bridges the connection between the NAR and the stochastic collocation approaches. It provides a unified way of understanding multi-fidelity models and allows for the following modifications to be made in order to combine the advantages of stochastic collocation and NAR:

- We introduce a noise-free assumption for the simulation data, which naturally extends our model to high-dimensional problems.
- By incorporating the NAR structure and an isotropic kernel function, predictions of the high-fidelity models no longer require extra simulations from the low-fidelity model.
- By incorporating a sequential learning approach to construct the multi-fidelity model, we propose Greedy-NAR, a Bayesian NAR that can construct itself automatically without the need of a special experimental design or *a priori* assumptions about the underlying physics.

We apply the sampling approach to several practical examples of the type found in energy engineering and science applications, demonstrating that it reduces the resultant surrogate model estimation error when dealing with limited computational resources for the high-fidelity data acquisition.

The paper is organized as follows. We firstly define the types of problems under consideration in Section 2. We then derive the model framework based on a general linear form in Section 3, followed by a discussion on the connections to stochastic collocation and NAR and our modifications for obtaining improved results. The experimental results are provided in Section 4.

2. Statement of the problem

We are interested in solutions to computational models that are parameter dependent, and where repeated evaluation of the model for different parameters is required. An archetypal example is a system of nonlinear steady-state or transient partial differential equations (PDEs) of arbitrary order for dependent scalar variables $u^i(\mathbf{x}, t; \boldsymbol{\xi})$, $i = 1, \ldots, I$, which can be collected in a vector function $\mathbf{u}$

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \mathcal{F}_{\boldsymbol{\xi}}(\mathbf{u}) = \mathcal{S}_{\boldsymbol{\xi}}(\mathbf{u}, \mathbf{x}, t), & (\mathbf{x}, t, \boldsymbol{\xi}) \in \Omega \times (0, T] \times \mathcal{X}, \\ \mathcal{B}_{\boldsymbol{\xi}}(\mathbf{u}) = 0, & (\mathbf{x}, t, \boldsymbol{\xi}) \in \partial\Omega \times (0, T] \times \mathcal{X}, \\ \mathbf{u}(\mathbf{x}, 0; \boldsymbol{\xi}) = \mathbf{u}_0(\mathbf{x}; \boldsymbol{\xi}), & (\mathbf{x}, t, \boldsymbol{\xi}) \in \Omega \times \{t = 0\} \times \mathcal{X}, \end{cases} \quad (1)$$

Here $\mathcal{F}_\xi$ is a spatial operator; S_ξ is a source term; Ω and $(0, T]$ are the spatial and temporal domain of interest; $\mathcal{B}_\xi$ is the boundary operator; and $\mathbf{u}_0(\mathbf{x}, \xi)$ is the initial condition. The variable $\mathbf{x} \in \Omega \subset \mathbb{R}^p$ is the spatial coordinate of a p-dimensional space and $t \in [0, T]$ is time. $\xi \in \mathcal{X} \subset \mathbb{R}^l$ is a vector of parameters that may appear in any number of $\mathcal{F}_\xi$, S_ξ, $\mathcal{B}_\xi$ and the initial condition $\mathbf{u}_0(\mathbf{x}; \xi)$. The PDEs can be nonlinear of any type. It is, however, assumed that the problem above is well posed (i.e., solutions always exist and are unique).

Eliminating the operators $\mathcal{F}_\xi$ and $\mathcal{B}_\xi$ leads to a system of ordinary differential equations (ODEs), which could include a molecular dynamics model, with $\mathbf{u}$ as the vector of the positions and velocities of all particles in the system. This is one of the examples we consider. We do not, however, exclude systems of algebraic equations. The method we develop is very general and can be applied to any parameter-dependent system.

A *high-fidelity model* is a numerical solver that solves for the $u^i(\mathbf{x}, t; \xi)$ such that the solutions $u^{i,H}(\mathbf{x}, t; \xi)$ are accurate approximations to $u^i(\mathbf{x}, t; \xi)$. The *quantity of interest* (QoI) can be concatenated vectorised solutions $u^{i,H}(\mathbf{x}, t; \xi)$ recorded at selected spatial-temporal locations, $\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}$, $t_1, \ldots, t_{n_t}$, or some scalar or vector quantity derived from $u^{i,H}(\mathbf{x}, t; \xi)$. The particular choice of the QoI does not affect our model, so we simply treat it as a function $\mathbf{y}^H(\xi) \in \mathbb{R}^d$ of the inputs, for some integer d, where for example $d = n_x \times n_t \times I$.

A *low-fidelity model* always exists, such that the solutions $u^{i,L}(\mathbf{x}, t; \xi)$ are less accurate than those of a high-fidelity solver but are cheaper to compute. For instance, we can use coarser spatial-temporal meshes; lower order time-stepping schemes, finite-difference stencils or basis expansions; or simplified physical models with fewer equations or with fewer terms. Let us denote the QoI derived from a low-fidelity model by $\mathbf{y}^L(\xi) \in \mathbb{R}^d$.

The issues we seek to address are:

- How can we deal with a QoI in a high-dimensional space (large d)?
- How can we accurately approximate the high-fidelity QoI $\mathbf{y}^H(\xi)$ by efficiently combining low- and high-fidelity solutions?
- Given limited computational resources, how can we design our experiments to achieve an optimal surrogate of the QoI? In particular, at which locations should we conduct our high-fidelity experiments to provide the maximum *information gain*?

3. Non-parametric Bayesian autoregression for multi-fidelity models

Let us consider a general linear form of autoregression for the multi-fidelity problem

$$\mathbf{y}^H(\xi) = \mathbf{W}\boldsymbol{\phi}(\mathbf{y}^L(\xi)) + \boldsymbol{\epsilon}^H, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{d \times l}$ is a weight matrix, $\boldsymbol{\phi}(\cdot)$ is an arbitrary feature mapping and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ is Gaussian noise. The feature mapping $\boldsymbol{\phi}(\cdot)$ can be to an infinite dimensional space ($l \to \infty$) in order to capture all of the essential features. To take account of the spatial correlations between entries of the weight matrix, we place a general matrix Gaussian prior over $\mathbf{W}$

$$\mathbf{W} \sim \mathcal{MN}(\mathbf{W}|\mathbf{0}, \mathbf{Q}, \mathbf{K}) = \frac{\exp\left(-\frac{1}{2}\mathrm{tr}[\mathbf{K}^{-1}\mathbf{W}^T\mathbf{Q}^{-1}\mathbf{W}]\right)}{(2\pi)^{dl/2}|\mathbf{Q}|^{l/2}|\mathbf{K}|^{d/2}}, \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^{l \times l}$ and $\mathbf{Q} \in \mathbb{R}^{d \times d}$ are the row and column covariance matrices. Working with an explicitly defined high-dimensional feature space is clearly impractical. Instead, we can integrate out $\mathbf{W}$ and introduce a kernel function to avoid specifying $\boldsymbol{\phi}$ and $\mathbf{K}$ explicitly, which leads to a probabilistic model for $\mathbf{y}^H(\xi)$ (derived in Appendix A). The marginal over $\mathbf{W}$ takes the form

$$\mathbf{y}^H(\xi) = \mathcal{MGP}(\mathbf{0}, k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi'))\mathbf{Q} + \sigma^2 \mathbf{I}). \tag{4}$$

in which $\mathcal{MGP}(\cdot, \cdot)$ denotes a multivariate GP, with mean function and covariance matrix given by the first and second arguments respectively.

$k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi'))$ is a general kernel. One common choice of kernel is the automatic relevance determination (ARD) kernel

$$k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi')) = \exp\left(-\sum_{m=1}^d \frac{|y_m^L(\xi) - y_m^L(\xi')|^2}{\theta_m^2}\right), \tag{5}$$

where $y_m^L(\xi)$ denotes the m-th element of $\mathbf{y}^L(\xi)$ and θ_m denotes the corresponding scaling factor, which determines the contribution of $y_m^L(\xi)$ to the approximation of $\mathbf{y}^H(\xi)$.

Assume that we have already collected simulation data $\mathbf{Y}^H = [\mathbf{y}_1^H, \ldots, \mathbf{y}_{N_H}^H]^T \in \mathbb{R}^{N_H \times d}$ and $\mathbf{Y}^L = [\mathbf{y}_1^L, \ldots, \mathbf{y}_{N_H}^L]^T \in \mathbb{R}^{N_H \times d}$, where we have introduced the notation $\mathbf{y}_i^H = \mathbf{y}^H(\xi_i)$ and $\mathbf{y}_i^L = \mathbf{y}^L(\xi_i)$, $i = 1, \ldots, N_H$, to aid the presentation. We can derive the log marginal likelihood $\mathcal{L}$ using model (4)

$$\mathcal{L} = \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathrm{vec}(\mathbf{Y}^H))^T\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}^H) - \frac{dN_H}{2}\ln(2\pi). \tag{6}$$

where $\boldsymbol{\Sigma} = \mathbf{Q} \otimes \mathbf{K}_H + \sigma^2\mathbf{I} \in \mathbb{R}^{dN_H \times dN_H}$ is the kernel matrix and the elements of $\mathbf{K}_H \in \mathbb{R}^{N_H \times N_H}$ are given by $K_{ij} = k^H(\mathbf{y}^L(\xi_i), \mathbf{y}^L(\xi_j))$. In this expression, $\otimes$ represents the Kronecker product. We can now use an optimization algorithm, e.g., L-BFGS-B, to obtain a maximum likelihood estimate (MLE) of the hyperparameters in the kernel function $k^H(\cdot, \cdot)$. The main computational cost is the inversion of $\boldsymbol{\Sigma}$, which is $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$ time and space complexity, respectively.

With the hyperparameters known, we can derive the posterior distribution for $\mathbf{y}^H(\xi)$, which is also a Gaussian with mean and covariance given by [22]

$$\mathbb{E}[\mathbf{y}^H(\xi)] = \left(\mathbf{Q} \otimes \mathbf{k}_*^H(\mathbf{y}^L(\xi))\right)^T\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}^H)$$
$$\mathrm{Var}[\mathbf{y}^H(\xi)] = \mathbf{Q}k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi))$$
$$- \left(\mathbf{Q} \otimes \mathbf{k}_*^H(\mathbf{y}^L(\xi))\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{Q} \otimes \mathbf{k}_*^H(\mathbf{y}^L(\xi))\right), \tag{7}$$

where $\mathbf{k}_*^H(\mathbf{y}^L(\xi)) = [k^H(\mathbf{y}^L(\xi), \mathbf{y}_1^L), \cdots, k^H(\mathbf{y}^L(\xi), \mathbf{y}_{N_H}^L)]^T$ is the correlation between the solution at any input ξ and the other observations in the training set.

3.1. Efficient computation of noise-free multivariate outputs

In the context of stochastic collocation [20] it is usually assumed that the noise term can be ignored (i.e., $\sigma \to 0$) because the low- and high-fidelity simulations are deterministic. In this case ($\sigma \to 0$), as shown in Appendix B, the predictive mean in Eq. (7) can be simplified (autokrigeability [1]), and our model can naturally handle high-dimensional outputs by setting $\mathbf{Q} = \mathbf{I}$ to obtain

$$\mathbf{y}^H(\xi) = \mathcal{MGP}\left(\mathbf{0}, k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi'))\mathbf{I}\right). \tag{8}$$

This compact GP model allows us to conduct efficient inference by the maximization of a compact likelihood

$$\mathcal{L}^H = \frac{d}{2}\ln|\mathbf{K}_H| - \frac{1}{2}\mathrm{tr}\left((\mathbf{Y}^H)^T\mathbf{K}_H^{-1}\mathbf{Y}^H\right) - \frac{N_H d}{2}\ln(2\pi). \tag{9}$$

The computational complexity for this new likelihood is $\mathcal{O}(N_H^3 d)$ rather than $\mathcal{O}(N_H^3 d^3)$, which allows us to apply the model to problems with a large d. If we substitute $\mathbf{Q} = \mathbf{I}$ into Eq. (7), we obtain the prediction

$$\mathbb{E}[\mathbf{y}^H(\xi)] = \left(\mathbf{k}_*^H(\mathbf{y}^L(\xi))\right)^T\boldsymbol{\Sigma}^{-1}\mathbf{Y}^H, \tag{10}$$

which has the same form as the predictions in the stochastic collocation framework [20].

Remark 1. We have introduced a Bayesian framework for the multi-fidelity problem in Eq. (5), which is a generalization of the stochastic collocation formulation. If we choose a noise free model (as in Eq. (8)) and a linear kernel, i.e., $k^H(\mathbf{y}^L(\xi)), \mathbf{y}^L(\xi')) = (\mathbf{y}^L(\xi))^T\mathbf{y}^L(\xi')$, we recover the classic stochastic collocation method in [20].

3.2. GP surrogate for low-fidelity models

In the original stochastic collocation framework [20], the prediction of a high-fidelity solution $\mathbf{y}^H(\xi)$ for an unseen simulation parameter ξ requires an execution of the low-fidelity model to obtain $\mathbf{y}^L(\xi)$. The underlying assumption is that the low- and high-fidelity correlation matrices are similar. This implies that the low-fidelity simulation cannot be significantly different from the high-fidelity simulation, which in turn implies that the low-fidelity model cannot be significantly cheaper.

Following the NAR model [21], we also recursively place a GP prior over the low-fidelity computer model to harness the function approximation utility of a GP. Specifically, we define

$$\mathbf{y}^L(\xi) = \mathcal{MGP}\Big(\mathbf{0}, k^L(\xi, \xi')\mathbf{I}\Big), \tag{11}$$

where $k^L(\cdot, \cdot)$ denotes the kernel function for the low-fidelity resolution. This model is not to be confused with (8) and here $k^L(\cdot, \cdot)$ is now a direct function of the inputs. Given some low-fidelity simulation data $\mathbf{Y}^L \in \mathbb{R}^{N_H \times d}$, we can obtain a marginal likelihood and posterior predictive process as before, yielding

$$\mathcal{L}^L = \frac{d}{2}\ln|\mathbf{K}_L| - \frac{1}{2}\text{tr}\big((\mathbf{Y}^L)^T\mathbf{K}_L^{-1}\mathbf{Y}^L\big) - \frac{Nd}{2}\ln(2\pi), \tag{12}$$

and [22]

$$\begin{aligned}
\mathbf{y}^L(\xi) &\sim \mathcal{N}(\boldsymbol{\mu}^L(\xi), \mathbf{\Pi}^L(\xi)) \\
\boldsymbol{\mu}^L(\xi) &= (\mathbf{k}_*^L(\xi))^T\mathbf{K}_L^{-1}\mathbf{Y}^L \\
\mathbf{\Pi}^L(\xi) &= \mathbf{I} \otimes \big(k^L(\xi, \xi) - (\mathbf{k}_*^L(\xi))^T\mathbf{K}_L^{-1}\mathbf{k}_*^L(\xi)\big),
\end{aligned} \tag{13}$$

in which the entries of $\mathbf{K}_L \in \mathbb{R}^{N_H \times N_H}$ are given by $k^L(\xi_i, \xi_j)$, and $\mathbf{k}_*^L(\xi) = [k^L(\xi, \xi_1), \cdots, k^L(\xi, \xi_{N_H})]^T$. Essentially, this recursive process produces a deep GP, which is infamous for its intractability; it requires expensive approximate inference, e.g., Monte-Carlo or variational Bayes. Fortunately, for the multi-fidelity problem the latent variable $\mathbf{y}^{(L)}(\xi)$ is observable by conducting low-fidelity simulations. This allows us to decompose the model training process through separable and independent likelihood functions (9) and (12).

Remark 2. By bridging the connection between stochastic collocation and NAR, we improve the stochastic collocation framework with NAR to provide an approximation to the low-fidelity model. Furthermore, we can take advantage of the additive structure [21] to strengthen the model by using a kernel function $k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi'))$ for Eq. (8).

3.3. High-fidelity model prediction

For model prediction corresponding to a unseen input ξ, if the low-fidelity solution is provided *via* low-fidelity simulation (as is done in the stochastic collocation literature [20]), we can easily compute the posterior mean using Eq. (19). The computational cost can be further reduced using our model, which is essentially a deep GP [5,6] with a trade-off in that we must sample from the distribution over $\mathbf{y}^L(\xi)$ to obtain a distribution over $\mathbf{y}^H(\xi)$. In detail, we first compute the posterior of $\mathbf{y}^L(\xi)$ *via* the Gaussian posterior (13). As is suggested in [7,21], we can generate I independent samples $\{\mathbf{y}^L(\xi^i)\}_{i=1}^I$ from $\mathcal{N}(\boldsymbol{\mu}^L(\xi), \mathbf{\Pi}^L(\xi))$ and approximate the posterior by

$$\begin{aligned}
p(\mathbf{y}^H(\mathbf{x})) &= \int (\mathbf{k}^L(\xi))^T\mathbf{K}_L\mathbf{Y}^L p(\mathbf{y}^L(\xi))d\mathbf{y}^L(\xi) \\
&\approx \sum_{i=1}^I \mathcal{N}\big(\boldsymbol{\mu}^H(\xi^i), \mathbf{\Pi}^H(\xi^i)\big).
\end{aligned} \tag{14}$$

3.4. Greedy selection of high-fidelity experiments

So far we have introduced our model based on the pre-existence of low- and high-fidelity data. In practice, such data (especially that of high-fidelity) is expensive to obtain. It is desirable to allocate computational resources, especially for the high-fidelity simulations, such that

the surrogate model can achieve its best performance with the least computational cost. Inspired by the work of [20], we intend to build the multi-fidelity surrogate model in a sequential manner.

Let us first consider the high-fidelity GP of (8). Assume that we already have a collection of model parameters $\{\xi_i\}_{i=1}^{N_H} \subset \mathcal{X}$ and the corresponding low- and high-fidelity simulation results $\mathbf{Y}^L$ and $\mathbf{Y}^H$. We define the *information gain* of a new model parameter ξ_* as the uncertainty or variance given the current data collection. Maximization of the information gain can then be defined mathematically by

$$\xi_* = \text{argmax}_{\xi \in \mathcal{X}}\big(\mathbf{k}_*^H(\mathbf{y}^L(\xi))\big)^T\mathbf{K}_H^{-1}\mathbf{k}_*^H(\mathbf{y}^L(\xi)), \tag{15}$$

where $\mathbf{k}_*^H(\mathbf{y}^L(\xi)) = [k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi_1)) \cdots, k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi_{N_H}))]^T$. In practice, however, Eq. (15) cannot be solved directly because $\mathbf{K}_H$ and $\mathbf{k}_*^H$ are unknown unless we already have a large number of high-fidelity simulations. The nonlinear kernel $k^h(\cdot, \cdot)$, e.g., ARD kernel, contains unknown hyperparameters whose estimation relies on the optimization of Eq. (9). In [20] this problem is addressed by implicitly assuming a linear kernel and a finite-cardinality subset of the input space $\mathcal{Z} \subset \mathcal{X}$ as the input candidates. The solution for Eq. (15) can then be found analytically by QR decomposition, LU decomposition, or Cholesky decomposition on $\mathbf{K}^H = (\mathbf{Y}^L)^T\mathbf{Y}^L$.

For the nonlinear kernel containing unknown hyperparameters, for which this method is not applicable, we develop an alternative approach. Essentially, in each step, we first update our high-fidelity GP model of Eq. (8) by maximizing the likelihood of Eq. (9). We then calculate the variance for all candidate locations except for those already chosen in previous steps. Next we conduct a high-fidelity experiment for the one that has the largest variance and add it to the data collection. This process is repeated until a maximum iteration number is reached.

Unlike the classic stochastic collocation model that requires low-fidelity simulations to provide correlation estimation, Greedy-NAR has the capacity to directly produce the low-fidelity simulations *via* the nonlinear kernel. This also allow us to apply the same process to construct the low-fidelity GP model in Eq. (11)

$$\xi_* = \text{argmax}_{\xi \in \mathcal{Z}}\big(\mathbf{k}_*^L(\xi)\big)^T\big(\mathbf{K}^L\big)^{-1}\mathbf{k}_*^L(\xi), \tag{16}$$

where $\mathbf{k}_*^L(\xi) = [k^H(\xi, \xi_1) \ldots, k^L(\xi, \xi_{N_H})]^T$. We present the full details of how to construct the model without requiring the prior execution of low-fidelity simulations for all candidates (which is required by the classic stochastic collocation model) in Algorithm 1[1].

The simplest way to allocate computational resources is by designing the number of executions for low- and high-fidelity simulations based upon the available computational budget. In case we want the system to be fully automatic, we may specify a large candidate set $\mathcal{Z}$ and the uncertainty bound (rather than specific number) for determining the number of iterations. We can also perform an eigenvalue analysis of the low-fidelity correlation matrix to find the optimal number of samples that fully capture the high-fidelity model behaviour within the parameter space.

Remark 3. Based on our unified framework for multi-fidelity models, we propose to equip our method with a sequential learning approach in order to improve efficiency. If we use a linear kernel for $k^H(\cdot, \cdot)$ for Algorithm 1, we recover the stochastic collocation sequential learning approach, which has a much simpler correlation structure (without hyperparameters).

3.5. Extension beyond bi-fidelity problems

So far we have only considered the bi-fidelity case. However, our method is readily extended to multi-fidelity problems. In particular, we

---

[1] The code is available as open-source on Github: ⟨12:monospace ⟩https://github.com/wayXing/GreedyNAR⟨/12:monospace⟩.
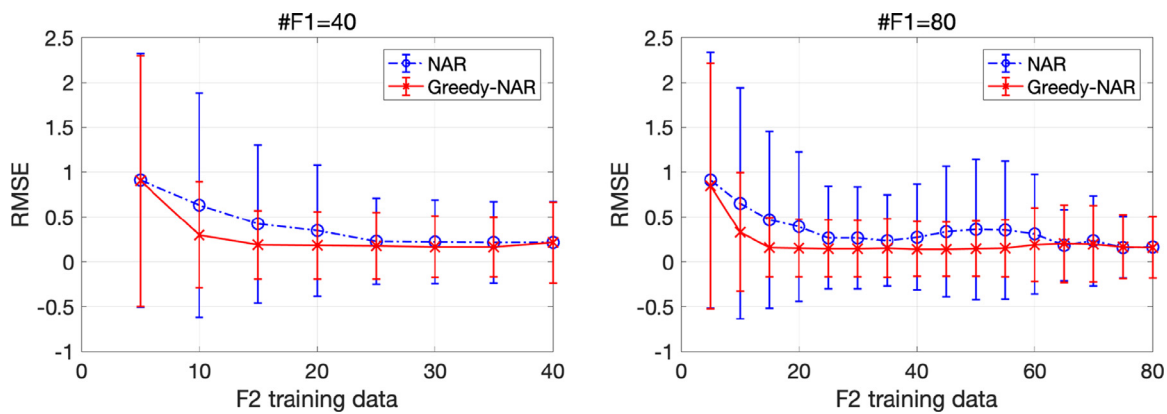
Fig. 1. Total RMSE of two-fidelity MD simulation QoIs, with comparisons to NAR.

Algorithm 1 Greedy-NAR construction

Input: A finite-cardinality set $\mathcal{Z}$ of input parameters, number of low- and high-fidelity experiments allowed N_L and N_H, initial data collection size N_0 ($N_0 < N_H < N_L$)

Output: A NAR model containing low- and high-fidelity GPs of Eq. 8 and 11

1: Randomly select N_0 point from $\mathcal{Z}$ and form the initial low-fidelity candidate set Γ^L
2: Conduct low-fidelity experiments for Γ_L and collect the solutions $\mathbf{Y}^L$
3: **for** $|\Gamma_L| < N_L$ **do**
4:     Update low-fidelity GP model by maximizing likelihood 12 based on Γ_L, $\mathbf{Y}^L$
5:     Find ξ_* based on Eq. 16 for $\mathcal{Z} \setminus \Gamma_L$
6:     Update low-fidelity candidate set $\Gamma_L \leftarrow \Gamma_L \cup \xi_*$
7:     Conduct a low-fidelity simulation to get solution $\mathbf{y}^L(\xi_*)$; update low-fidelity solution set $\mathbf{Y}_L \leftarrow [\mathbf{Y}_L^T, (\mathbf{y}^L(\xi_*))^T]^T$
8: **end for**
9: Randomly select N_0 point from Γ_L and form initial high-fidelity candidate set Γ_H
10: Conduct low-fidelity experiments for Γ_H and collect the solutions $\mathbf{Y}^H$
11: **for** $|\Gamma_H| < N_H$ **do**
12:     Update high-fidelity GP model by maximizing likelihood 9 based on Γ_H, $\mathbf{Y}^H$
13:     Find ξ_* beasd on Eq. 15 for $\Gamma_L \setminus \Gamma_H$
14:     Update high-fidelity candidate set $\Gamma_H \leftarrow \Gamma_H \cup \xi_*$
15:     Conduct a high-fidelity simulation to get solution $\mathbf{y}^H(\xi_*)$; Update low-fidelity solution set $\mathbf{Y}_H \leftarrow [\mathbf{Y}_H^T, (\mathbf{y}^H(\xi_*))^T]^T$
16: **end for**

---

use Eq. (11) to predict the second fidelity and recursively use Eq. (8) to pass down the predictions in the deep GP structure to derive the final high-fidelity predictions. For the greedy construction of Greedy-NAR in the multi-fidelity problem, we can simply add Eq. (8) after each new fidelity step and optimize Eq. (15), in which 'low-fidelity' now refers to the fidelity level of the preceding step and 'high-fidelity' is the next fidelity level in the chain.

4. Results and discussion

4.1. Test problem 1: Molecular dynamics simulation model

We first consider a molecular dynamics (MD) simulation using a Lennard-Jones (LJ) potential, with between 36 and 696 particles of ^{12}C in a cubic simulation box with periodic boundary conditions. Details of the implementation are provided in Appendix C.

We chose the low-fidelity simulation model as one with a large-time step, and ran it on randomly selected points in the parameter space, consisting of temperature and density. The low-fidelity simulations used a time step of 10 or 20 fs and the high-fidelity simulations used a time step of 1 fs. For all model parameter values, the QoIs were computed using the low-fidelity model. These QoIs were in the form of either scalars or vectors, namely the radial distribution function (RDF), mean squared displacement (MSD), averaged total energy and self-diffusion coefficient.

All training and test data were normalized using $y_{nd} \mapsto (y_{nd} - \bar{y}_d)/\sigma_d^2$, where y_{nd} denotes the n-th sample of the dth coefficient of an output; $\bar{y}_d = (1/M) \sum_{n=1}^{M} y_{nd}$ is the mean value of the dth coefficient over M samples, and $\sigma_d^2 = \sum_{i=1}^{M} (y_{nd} - \bar{y}_d)/(M-1)$ is the empirical variance. This normalizing process was applied as a pre-processing step for all experimental data in all examples unless stated otherwise.

We ran 40 and 80 low-fidelity simulations (we call the low-fidelity F1 from here on) at randomly selected simulation parameters, based on which we slowly increased the number of training data points at the second fidelity level (F2). The performance was evaluated based on the root mean square error (RMSE) on 34 test points. The experiments were repeated five times with random shuffling of training and test data, and the errors were averaged to obtain the final error.

The averaged root mean square error (RMSE) statistics (mean and standard deviation) with 40 and 80 F1 training data points and an increasing number of F2 training points are shown in Fig. 1, along with the equivalent results for NAR. These RMSE errors are the sums of the RMSE errors of all QoIs. The trends for each individual QoI RMSE were the same, so only the total RMSE is shown unless stated otherwise. Greedy-NAR results in a significantly lower RMSE and a smaller variance when the F2 data is limited (i.e., #F2 = 10). Both methods converge to the same RMSE mean and standard deviation as the number of F2 points is increased (indicating that Greedy-NAR indeed converges to the standard NAR).

We then evaluated Greedy-NAR in a more complicated case in which we aim to find an accurate approximation of the QoI function from the highest fidelity output (F3) in a 3-fidelity simulation. We show the RMSE as a function of F3 training data points given different numbers of F1 and F2 training data points in Fig. 2. The experiments were again conducted five times with random shuffling of training and test data.

It is clear that with a higher fidelity example, the performance gain over NAR increases substantially. For example, with only 15 F3, 20 F2, and 40 F1 training data points, Greedy-NAR achieves a similar performance to the standard NAR with 80 F1, F2, and F3 training data points. Note that for #F1=80 and #F2=40, Greedy-NAR converges to the lowest RMSE out of all combinations of training data points, which indicates that increasing the number of training data points at different fidelities does not guarantee a performance improvement, due to overfitting.
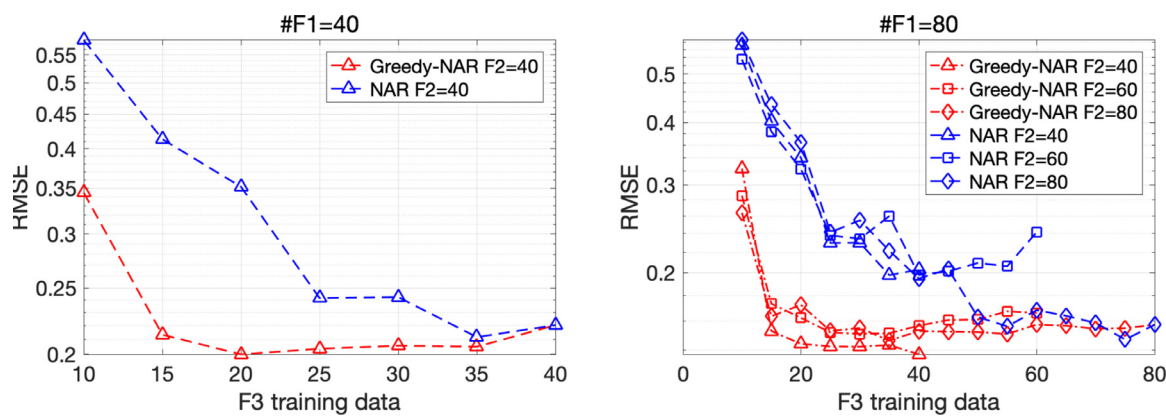
Fig. 2. Total RMSE of the three-fidelity MD simulation QoIs with comparisons to NAR.
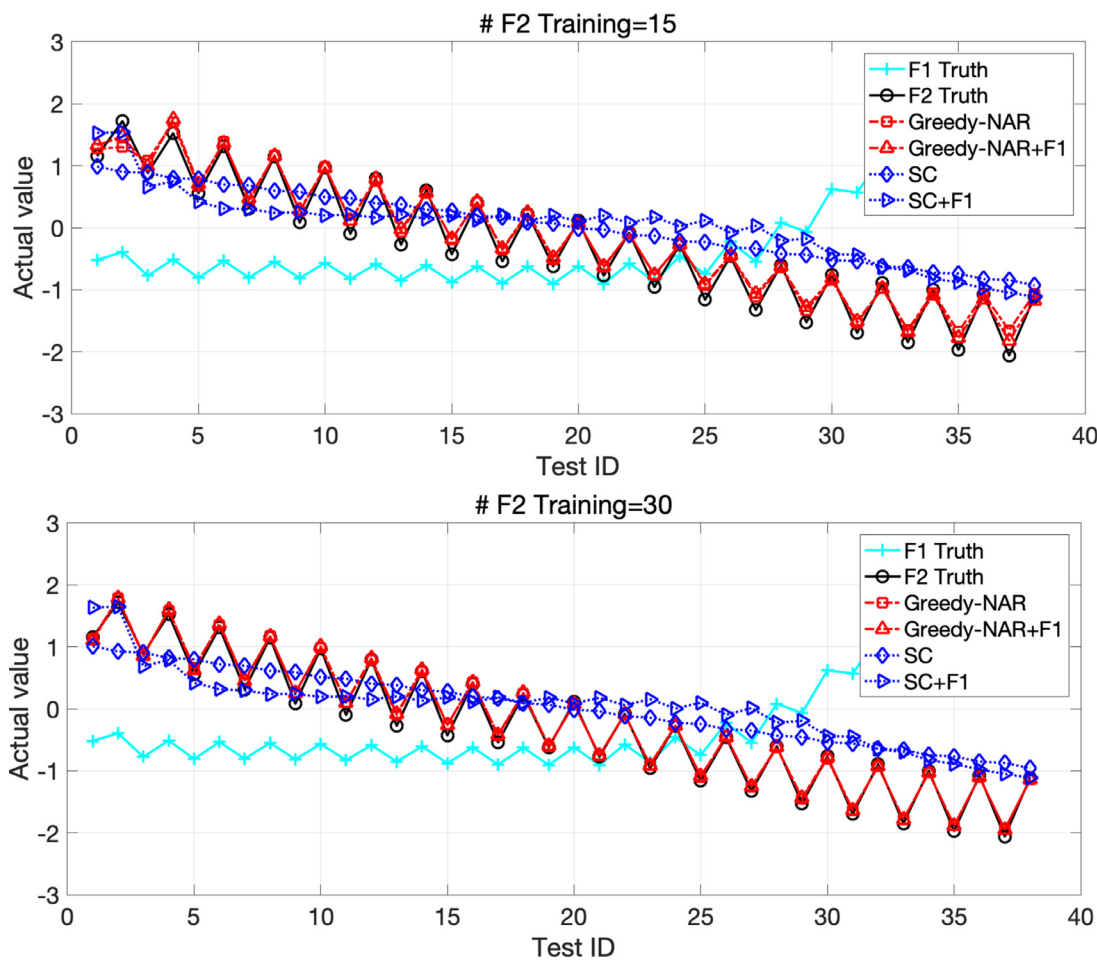

Fig. 3. Averaged total energy predictions of a two-fidelity MD with comparisons to SC.

Nevertheless, Greedy-NAR exhibits dramatic improvements over standard NAR in terms of the rate of convergence of the error, especially with an increasing number of F3 training data points, suggesting a huge computational saving in practical scenarios.

Finally, we compared Greedy-NAR with stochastic collocation (SC) in the bi-fidelity (F1 and F2) case. Since a standard stochastic collocation requires F1 observations as test points to give correct F2 predictions, we also supply F2 predictions to both models as comparisons. The test points are sorted such that their corresponding inputs are arranged in increasing order for the purposes of visualization. The averaged total energy and self-diffusion coefficient predictions are shown in Figs. 3 and 4, respectively. As is evident from these figures, the F1 and F2 simulation results (ground truths) for the RDF exhibit quite different trends, indicating different correlation structures for the F1 and F2 target functions. Since SC relies on the assumption that the correlations of different fidelities are similar, it is not surprising to see that it fails to give accurate predictions even given the F1 test point values. In contrast, Greedy-NAR performs very well even without the F1 values; with the F1 test point values provided, the predictive accuracy of Greedy-NAR further improves.
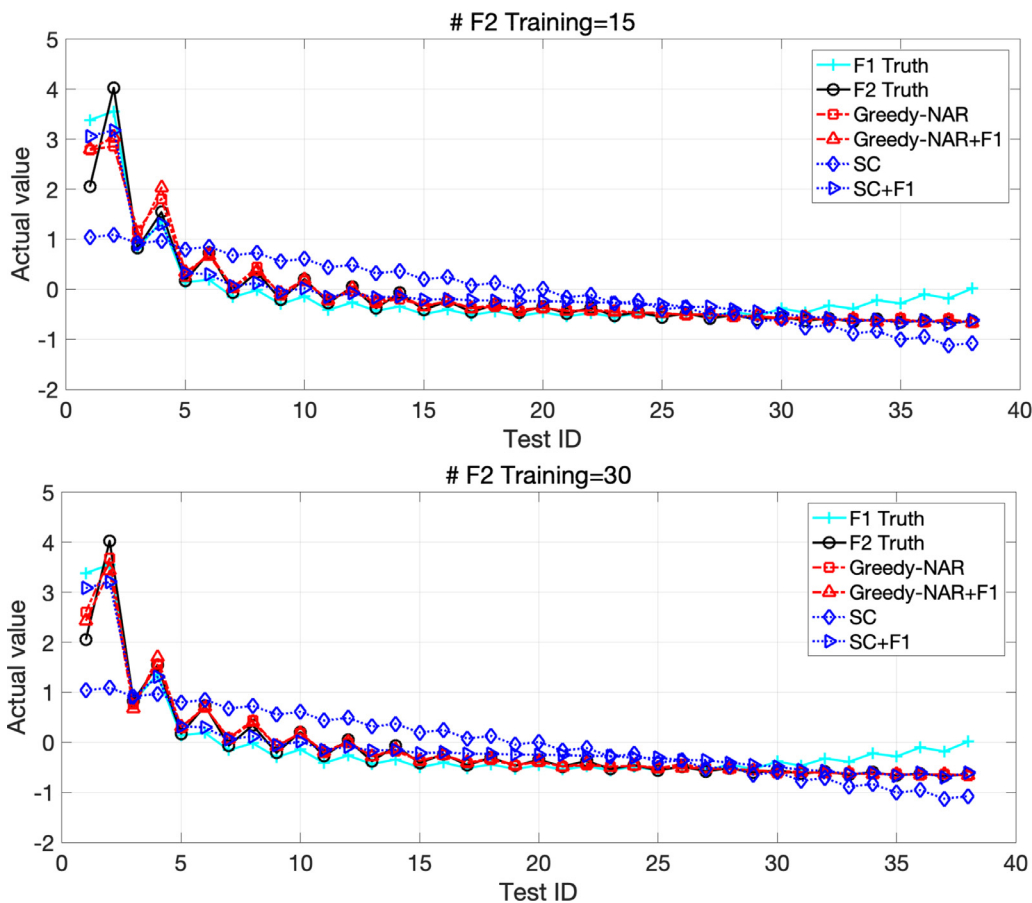
Fig. 4. Self-diffusion coefficient predictions of a two-fidelity MD with comparisons to SC.
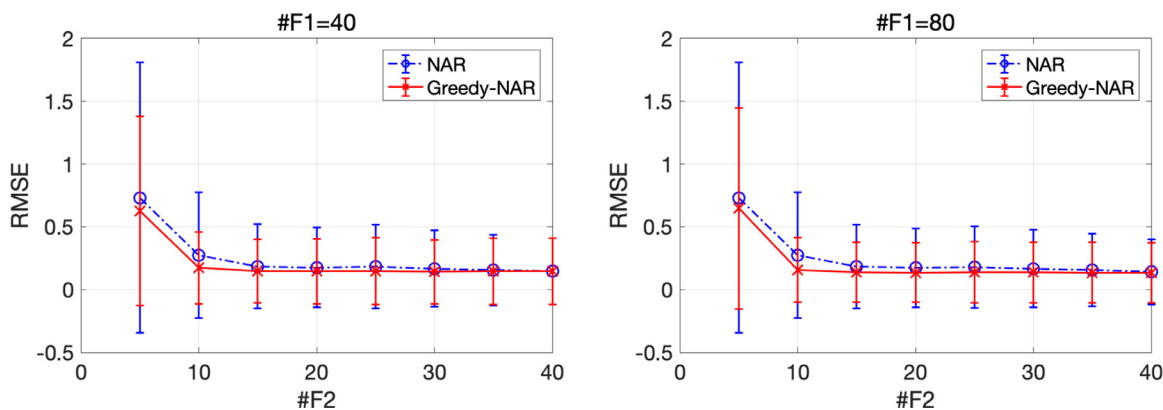

Fig. 5. Total RMSE of the bi-fidelity plasmonic nano-particle simulation with comparisons to NAR.

4.2. Test Problem 2: Plasmonic nano-particle arrays

In this test problem, we calculate the extinction and scattering efficiencies Q_{ext} and Q_{sc} using the Coupled Dipole Approximation (CDA) method for plasmonic systems with different numbers of scatterers. CDA is an approach for simulating the optical response of an array of identical non-magnetic metallic nano-particles with sizes much smaller than the wavelength of light (here 25 nm).

Q_{ext} and Q_{sc} are defined here as the QoIs. We used our proposed approach to construct surrogate models for the efficiencies with up to three fidelities. We considered particle arrays given by Vogel spirals. The number of nano-particles in a plasmonic array significantly impacts the local extinction field induced by plasmonic arrays since the number

of interactions of incident waves from particles influences the magnetic field. Fig. 13 shows configurations of Vogel spirals with particle numbers in the set {2, 25, 50}, which define fidelities F1 to F3. The parameter space was taken to be $\lambda \in [200, 800]$ nm, $\alpha_{vs} \in [0, 2\pi]$ rad, and $a_{vs} \in (1, 1500)$. These are the incident wavelength, the divergence angle and scaling factor, respectively. Inputs were selected using a Sobol sequence. The computational time needed for running CDA grows exponentially with the number of nano-particles. Therefore, the computational savings associated with the proposed sampling strategy are significant. Details of the model and numerical implementation are provided in Appendix C.

Similar to the previous example, we evaluated the sum of the RMSEs of the QoIs as a function of F2 training point number on 684 test
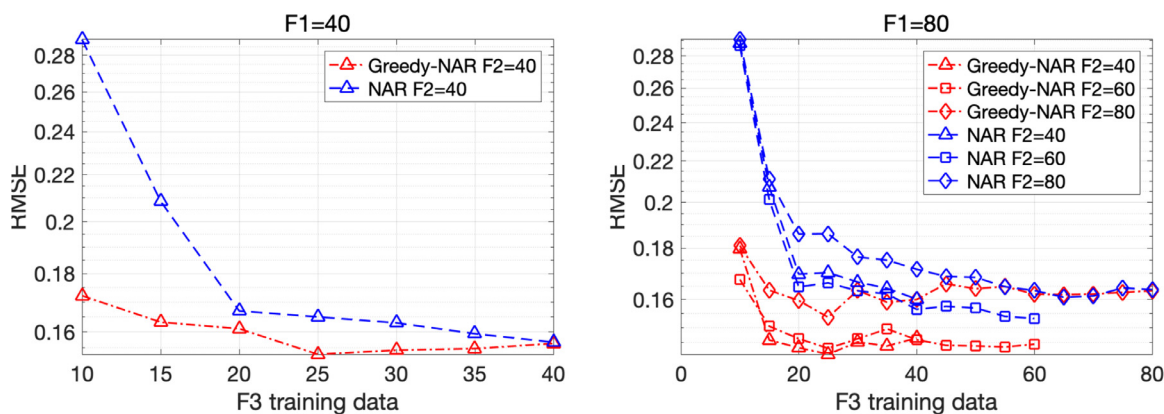
Fig. 6. Total RMSE of the three-fidelity plasmonic nano-particle simulation with comparisons to NAR.
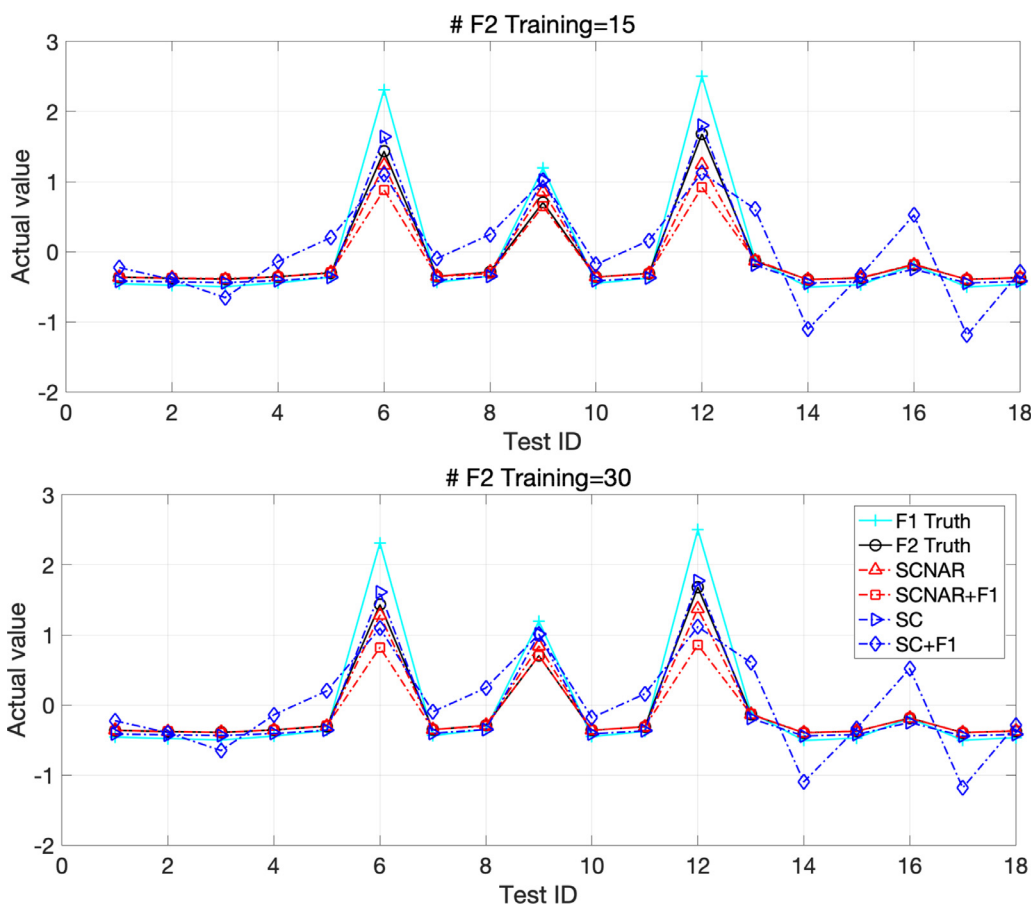

Fig. 7. Extinction coefficient predictions from the plasmonic nano-particle simulations with comparisons to SC.

points when given 40 and 80 F1 training samples. The experiment was repeated five times with random shuffling of training and test data. The RMSE statistics (mean and standard deviation) with 40 and 80 F1 training data points are shown in Fig. 5. Although in this experiment Greedy-NAR does not provide any significant advantage over NAR, it still shows consistent superiority and convergence towards the standard NAR result. Greedy-NAR demonstrates its value in terms of simulation time because the extra computational cost required by Greedy-NAR is negligible compared with the expense of running high-fidelity simulations.

We then evaluated Greedy-NAR and NAR in a 3-fidelity case. The performances of both methods given different combinations of F1, F2, and F3 training data points are given in Fig. 6, again with 684 test points.

We obtain results that are similar to those of the previous example, in that Greedy-NAR substantially outperforms standard NAR, particularly when the number of training samples is low. In this experiment, we can also see that with 80 F2 training data points, the performances of both methods is worse than that with 40 and 60 F2 training data points, which is a clear symptom of overfitting.

Finally, we compared our method with stochastic collocation, and the results are provided in Figs. 7 and 8. It is clear that for both the extinction and scattering coefficients, stochastic collocation without F1 test points as model inputs is inaccurate, deviating significantly from the ground-truth values. However, it provides accurate predictions when the F1 test points are revealed to the model. Given the F1 test points,
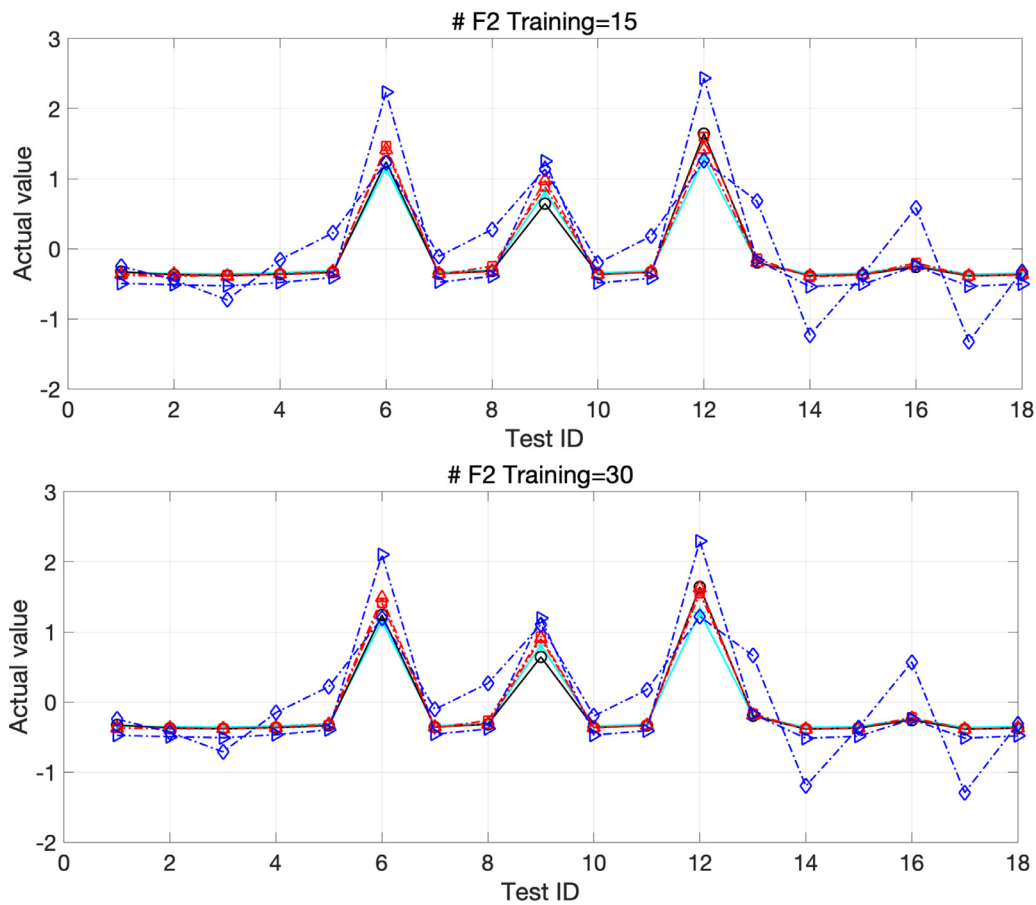
Fig. 8. Scattering coefficient predictions from the plasmonic nano-particle simulations with comparisons to SC.
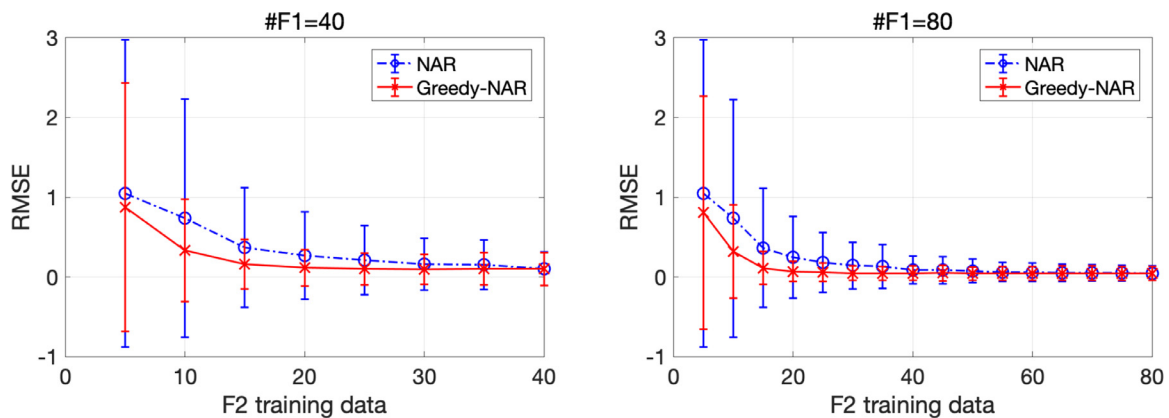

Fig. 9. Total RMSE of the two-fidelity mixing flow simulation, with comparisons to NAR.

stochastic collocation achieves the best performance for some special cases, e.g., test ID 12 in Fig. 7. For the extinction coefficient, SC with F1 test points shows a similar performance to Greedy-NAR with F1 test points and Greedy-NAR. Overall, we conclude that with enough F1 test points revealed to the model, stochastic collocation is a good method when the F1 and F2 ground-truth functions have a similar correlation structure. Otherwise, SC with or without the F1 test points generally provides inaccurate predictions. We also note that in most practical cases, providing F1 test data means running expensive simulations, which is often prohibitive.

4.3. Test Problem 3: mixing flow in an elbow-shape pipe

When it comes to modeling turbulent flows in different geometrical configurations, there is a large set of available models ranging from simple one-equation models such as Sparllat–Almaras to sophisticated models such as the Large Eddy Simulation (LES). The computational models available can be classified as low- or high-fidelity models, or classified using more fidelity levels. A major benefit of using low-fidelity models arises when they are employed in the context of design and optimization of thermal-fluid systems. In these scenarios, the
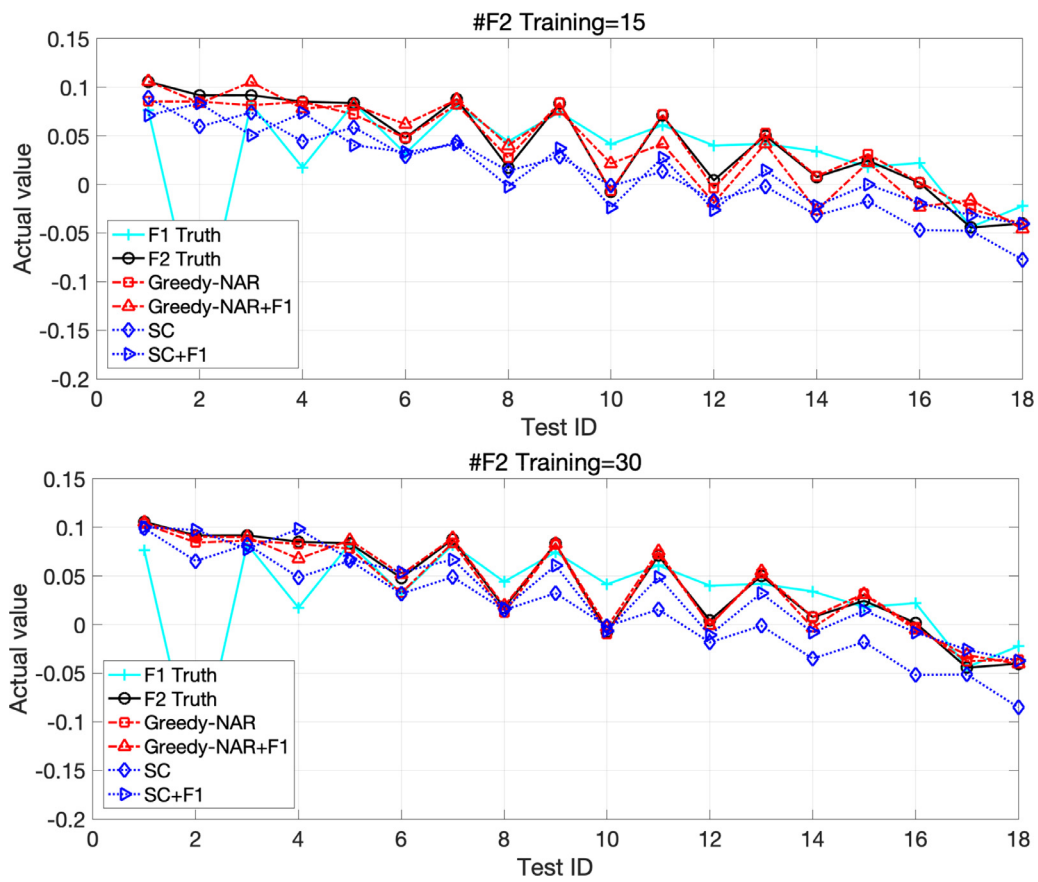
Fig. 10. The F2 4th coefficient predictions for the mixing flow problem, with comparisons to NAR.
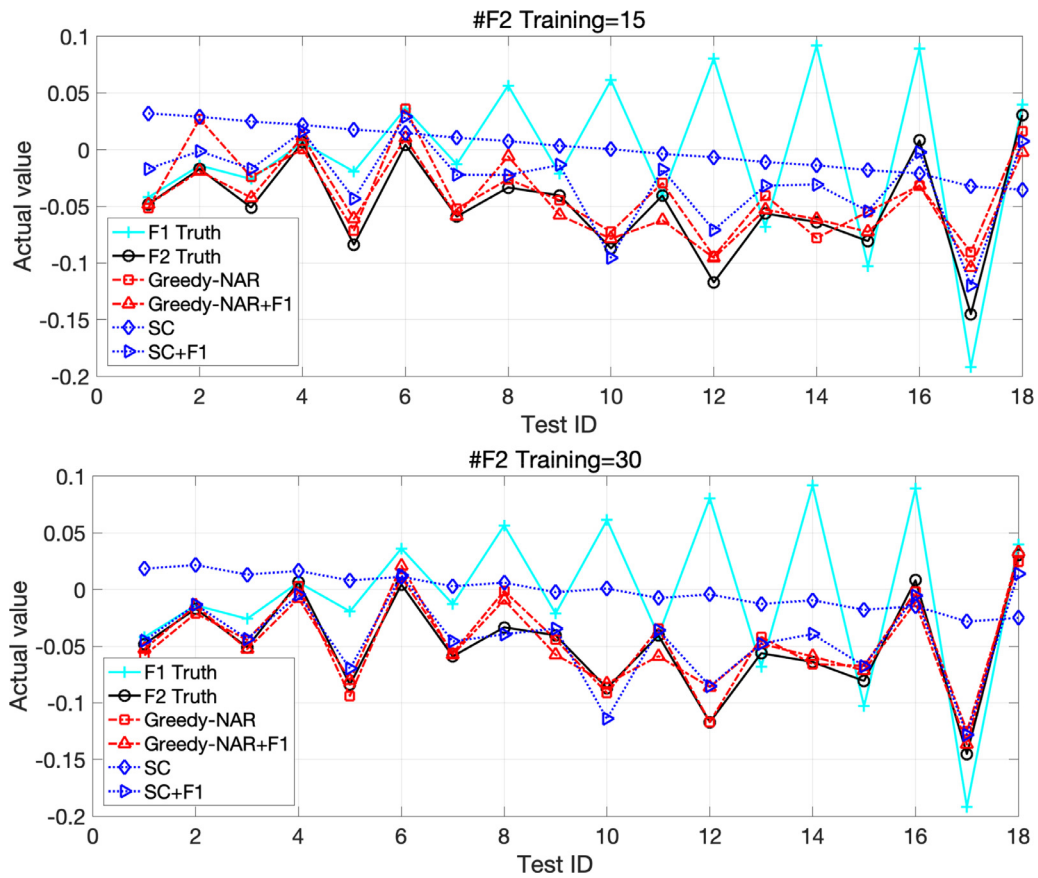

Fig. 11. The F2 tth coefficient predictions for the mixing flow problem, with comparisons to NAR.
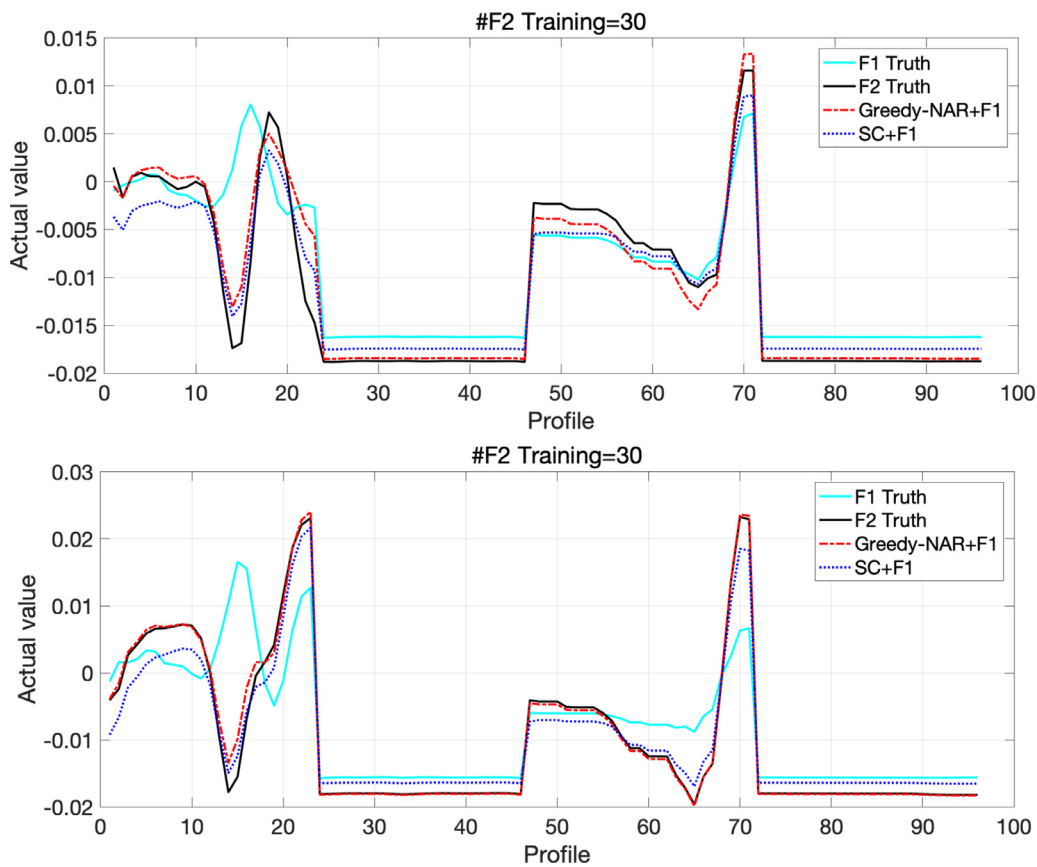
Fig. 12. Two actual F2 profile predictions of mixing flow.

computational speed of these models benefits exploration of the design space. However, their inherent inaccuracy may lead to sub-optimal designs. Fusion of low- and high-fidelity models is believed to be an effective way of balancing computational costs against the accuracy of predictions.

We applied our Greedy-NAR to a benchmark problem of mixing flow in a pipe. The input parameter space in this case was the freestream velocity at the large inlet and the freestream velocity at the smaller inlet. The QoIs were two (vectorised) velocity and pressure profiles in the symmetry plane of the elbow pipe, one at the elbow junction and the other near the pipe exit, both at $t = 50$ seconds. The vector of these quantities had 96 elements. The F1 model was the Sparlat–Allmaras turbulence model, while the F2 model was the LES model. Details of the numerical implementation are provided in Appendix C.

We first ran 40 and 80 low-fidelity simulations at randomly selected parameter values, based on which we gradually increased the number of F2 training samples. As before we used the total RMSE of the QoIs (for 18 test points) averaged from the results of a 5-fold cross validation to assess the results (Fig. 9). The RSME for Greedy-NAR is significantly lower, with a smaller variance when the F2 data is limited (e.g., to 10). Both methods converge to the same performance level in terms of error mean and standard deviation.

Both methods provide very accurate predictions for the 1st, 2nd, and 3rd coefficients in the expansion (24) (basis coefficients in a reduced-dimensional space for the outputs **y**, as described in Appendix C) since all these values, which represents low-frequency components of the data, are very stable. The predictions of the 4th and 5th coefficients are shown in Figs. 10 and 11. We can draw the same conclusion as before, that Greedy-NAR achieves the most accurate results, especially with the F1 test points observations revealed. In contrast, SC without F1 observations invariably leads to poor performance.

To demonstrate the accuracy on high-dimensional predictions, we show two test cases, based on 80 F1 and 30 F2 training samples, in Fig. 12. The actual values are normalized for visualization. Greedy-NAR leads to very accurate predictions, clearly close to the F2 ground truths, whereas SC makes good predictions overall except for areas where the F1 and F2 ground truths exhibit different trends.

5. Summary and Conclusions

In this paper we propose a uniform Bayesian framework that connects the classical methods of multi-fidelity modelling. Greedy-NAR is a nonlinear Bayesian autoregressive model that captures complex between-fidelity correlations but is constructed in a way that it outperforms the standard methods when the computational budget is limited.

We demonstrated through a variety of problems that Greedy-NAR leads to improvements in both the stability and accuracy of predictions compared to NAR. Comparisons to SC demonstrated advantages in terms of both accuracy and computational cost. It is, moreover, easy to implement and highly scalable, taking full advantage of low-fidelity results and very effectively limiting the number of high-fidelity simulations that need to be performed for a given accuracy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Appendix A. Kernel formulation of generalised autoregression model

We first integrate out $\mathbf{W}$ in Eq. (2)

$$
\begin{aligned}
p(\mathbf{y}^H(\xi)) &= \int p(\mathbf{y}^H(\xi) \mid \mathbf{W}, \mathbf{y}^L(\xi)) p(\text{vec}(\mathbf{W})) \, d\text{vec}(\mathbf{W}) \\
&= \int \mathcal{N}(\mathbf{y}^H(\xi) \mid \mathbf{W}\boldsymbol{\phi}(\mathbf{y}^L(\xi)), \sigma^2\mathbf{I}) \mathcal{N}(\text{vec}(\mathbf{W}) \mid \mathbf{0}, \mathbf{K} \otimes \mathbf{Q}) \, d\text{vec}(\mathbf{W}) \\
&= \int \mathcal{N}(\mathbf{y}^H(\xi) \mid (\boldsymbol{\phi}(\mathbf{y}^L(\xi)) \otimes \mathbf{I})^T \text{vec}(\mathbf{W}), \sigma^2\mathbf{I}) \mathcal{N}(\text{vec}(\mathbf{W}) \mid \mathbf{0}, \mathbf{K} \otimes \mathbf{Q}) \, d\text{vec}(\mathbf{W}) \\
&= \mathcal{N}(\boldsymbol{\xi}_n \mid \mathbf{0}, (\boldsymbol{\phi}(\mathbf{y}^L(\xi)) \otimes \mathbf{I})^T (\mathbf{K} \otimes \mathbf{Q})(\boldsymbol{\phi}(\mathbf{y}^L(\xi)) \otimes \mathbf{I}) + \sigma^2\mathbf{I}) \\
&= \mathcal{N}(\boldsymbol{\xi}_n \mid \mathbf{0}, \boldsymbol{\phi}^T(\mathbf{y}^L(\xi))\mathbf{K}\boldsymbol{\phi}(\mathbf{y}^L(\xi))\mathbf{Q} + \sigma^2\mathbf{I}).
\end{aligned}
\tag{17}
$$

noting that $\text{vec}(\mathbf{W}) \sim \mathcal{N}(\text{vec}(\mathbf{W}) \mid \mathbf{0}, \mathbf{K} \otimes \mathbf{Q})$, where $\text{vec}(\mathbf{W}) = (\mathbf{w}_1^T, \ldots, \mathbf{w}_l^T)^T \in \mathbb{R}^{lf}$ is the vectorization of $\mathbf{W}$. In this equation $\otimes$ is the Kronecker product. Since $\mathbf{K}$ is symmetric and positive semidefinite (PSD) by definition, it possesses a unique PSD symmetric square root $\sqrt{\mathbf{K}}$. Hence the term $\boldsymbol{\phi}^T(\mathbf{y}^L(\xi))\mathbf{K}\boldsymbol{\phi}(\mathbf{y}^L(\xi))$ defines an inner product $\langle \cdot, \cdot \rangle_{\mathbf{K}}$ in the feature space

$$
\langle \boldsymbol{\phi}(\mathbf{y}^L(\xi)), \boldsymbol{\phi}^T(\mathbf{y}^L(\xi')) \rangle_{\mathbf{K}} = \boldsymbol{\phi}^T(\mathbf{y}^L(\xi))\mathbf{K}\boldsymbol{\phi}(\mathbf{y}^L(\xi')) = \langle \tilde{\boldsymbol{\phi}}(\mathbf{y}^L(\xi)), \tilde{\boldsymbol{\phi}}(\mathbf{y}^L(\xi')) \rangle
\tag{18}
$$

for $\xi, \xi' \in \mathcal{X}$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product and $\tilde{\boldsymbol{\phi}}(\mathbf{y}^L(\xi)) = \sqrt{\mathbf{K}}\boldsymbol{\phi}(\mathbf{y}^L(\xi))$. Eq. (17) defines a multivariate GP with a kernel given by the inner product $\langle \cdot, \cdot \rangle_{\mathbf{K}}$. We can now employ kernel substitution [27], which consists of replacing the kernel in (18) with a general kernel $k^H(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi'))$.

Appendix B. Autokrigeability

For $\sigma \to 0$, the predictive mean in Eq. (7) can be simplified as follows:

$$
\begin{aligned}
\mathbb{E}[\mathbf{y}^H(\xi)] &= (\mathbf{Q} \otimes \mathbf{k}_*^H(\mathbf{y}^L(\xi)))^T (\mathbf{Q} \otimes \mathbf{K}_H)^{-1} \text{vec}(\mathbf{Y}^H) \\
&= \mathbf{Q} \otimes (\mathbf{k}_*^H(\mathbf{y}^L(\xi)))^T (\mathbf{Q}^{-1} \otimes \mathbf{K}_H^{-1}) \text{vec}(\mathbf{Y}^H) \\
&= (\mathbf{Q}\mathbf{Q}^{-1}) \otimes (\mathbf{k}_*^H(\mathbf{y}^L(\xi)))^T \mathbf{K}_H^{-1} \text{vec}(\mathbf{Y}^H) \\
&= \mathbf{I} \otimes (\mathbf{k}_*^H(\mathbf{y}^L(\xi)))^T \mathbf{K}_H^{-1} \text{vec}(\mathbf{Y}^H). \\
&= \text{vec}\left((\mathbf{k}_*^H(\mathbf{y}^L(\xi)))^T \mathbf{K}_H^{-1} \mathbf{Y}^H \mathbf{I}^T\right) \\
&= (\mathbf{Y}^H)^T \mathbf{K}_H^{-1} \mathbf{k}_*^H(\mathbf{y}^L(\xi))
\end{aligned}
\tag{19}
$$

This is also known as autokrigeability [1], and it reveals that the actual structure and value of $\mathbf{Q}$ do not matter as far as the mean predictions are concerned because $\mathbf{Q}$ will always cancel. By assuming noise-free observations, our model can thus naturally handle high-dimensional outputs without the complex computations associated with $\mathbf{Q}$. For simplicity, therefore, we can set $\mathbf{Q} = \mathbf{I}$ to obtain the model (8).

Appendix C. Details of the numerical experiments

Molecular dynamics model

The interatomic interactions in molecular dynamics (MD) are defined using a potential function. In this example we used the Lennard-Jones (LJ) potential [19]

$$
u(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^6 \right],
\tag{20}
$$

where, r_{ij} denotes the pairwise distance between particles i and j, ϵ is the potential well depth and σ defines the length scale for this pairwise interatomic interaction model. When the size of the integration time-step is too large, the numerical scheme can become highly unstable. In order to prevent such numerical instability, we capped the magnitude of the repulsive interactions for closely approaching atoms when the ratio

of σ/r_{ij} exceeded 1.2. This force capping is essential in many problems and is described in detail in [18].

We assumed σ and ϵ to be 3 **r**A and 1 kcal mol^{-1}, respectively. Periodic boundary conditions were used on all sides of the cubic simulation box, with width $L = 27.05$ **r**A. We considered a uniform grid of temperature T and density ρ defined as $T \times \rho$: [500, 1000]K $\times$ [36.27, 701.29]kg m^{-3} with 114 inputs selected from this grid. The density range was equivalent to $\rho^* \in [0.05, 0.95]$, where ρ^* is the dimensionless density equal to $N\sigma^3/V$ ($\rho = Nm/V$), in which m is the particle mass, set to $m = 12.01$ g mol^{-1}, and $V = L^3$ is the simulation box volume. Based on the dimensionless density range, the number of molecules ranged from 36 to 696. All simulations were performed using the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) code. The Verlet algorithm was employed for integration of the equations of motion and a microcanonical (NVE) ensemble was assumed.

Plasmonic array model

The response of a plasmonic array to electromagnetic radiation can be computed by the solution of the local electric fields, $\mathbf{E}_{loc}(\mathbf{r}_j)$, for each of the nano-spheres. Considering N metallic particles described by the same volumetric polarizability $\alpha(\omega)$ and located at vector positions $\mathbf{r}_i$, the local field $\mathbf{E}_{loc}(\mathbf{r}_j)$ can be computed by solving [9]

$$
\mathbf{E}_{loc}(\mathbf{r}_i) = \mathbf{E}_0(\mathbf{r}_i) - \frac{\alpha k^2}{\epsilon_0} \sum_{j=1, j \neq i}^N \tilde{\mathbf{G}}_{ij} \mathbf{E}_{loc}(\mathbf{r}_j)
\tag{21}
$$

where $\mathbf{E}_0(\mathbf{r}_i)$ is the incident field, k is the wavenumber in the background medium, ϵ_0 is the dielectric permittivity of vacuum ($\epsilon_0 = 1$ in the CGS unit system), and $\tilde{\mathbf{G}}_{ij}$ is constructed from 3×3 blocks of the overall $3N \times 3N$ Green's matrices for the ith and jth particles. $\tilde{\mathbf{G}}_{ij}$ is a zero matrix when $j = i$, and is otherwise computed as

$$
\tilde{\mathbf{G}}_{ij} = \frac{\exp(ikr_{ij})}{r_{ij}} \left\{ \mathbf{I} - \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}^T - \left[\frac{1}{ikr_{ij}} + \frac{1}{(kr_{ij})^2}(\mathbf{I} - 3\hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}^T) \right] \right\}
\tag{22}
$$

where $\hat{\mathbf{r}}_{ij}$ is the unit position vector from particles j to i and $r_{ij} = |\mathbf{r}_{ij}|$. By solving Eqs. (21) and (22), the total local fields $\mathbf{E}_{loc}(\mathbf{r}_i)$, and as a result the scattering and extinction cross-sections, are computed. Details of the numerical solution can be found in [25].

Q_{ext} and Q_{sc} are obtained by normalization of the scattering and extinction cross-sections with respect to the total projected area of the array. We considered a class of particle arrays known as Vogel spirals, which are defined by [3]

$$
\rho_n = \sqrt{n}a_{vs} \quad \text{and} \quad \theta_n = n\alpha_{vs},
\tag{23}
$$

where ρ_n and θ_n denote, respectively, the radial distance and the polar angle of the nth particle in a Vogel spiral array. Hence, the incident wavelength λ, the divergence angle α_{vs}, the scaling factor a_{vs}, and the number of particles n can uniquely define the Vogel spiral configuration.

Turbulent mixing model

As illustrated in Fig. 14, water flows from two inlets, the bottom left end of the pipe and a smaller inlet located on the elbow. The water exits the pipe from the top right in a horizontal upward direction. The freestream velocity at the large inlet (with a diameter of 1 m) varies between 0.2 to 2 m s^{-1}, and the freestream velocity at the smaller inlet (with a diameter 0.5 m), varies between 1.2 to 3 m s^{-1}. The fluid density was taken to be 1000 kg m^{-3} and the viscosity to be 8×10^{-4} Pa s. We ran Sparlart–Allmaras in unsteady mode just as a benchmark. The algorithms LES (F2) and Sparlart–Allmaras (F1) were implemented in ANSYS Fluent, employing the dynamic kinetic Energy subgrid-scale model within the LES model. The default was used, namely a second-order implicit formulation and central differencing (finite volume). For Sparlart–Allmaras, the default vorticity-based production and low-Reynold's number damping were selected. For both F1 and F2 simulations, a total of 36,134 nodes and 29,399 hexahedral cells were used, along with a time step of 0.01 s.
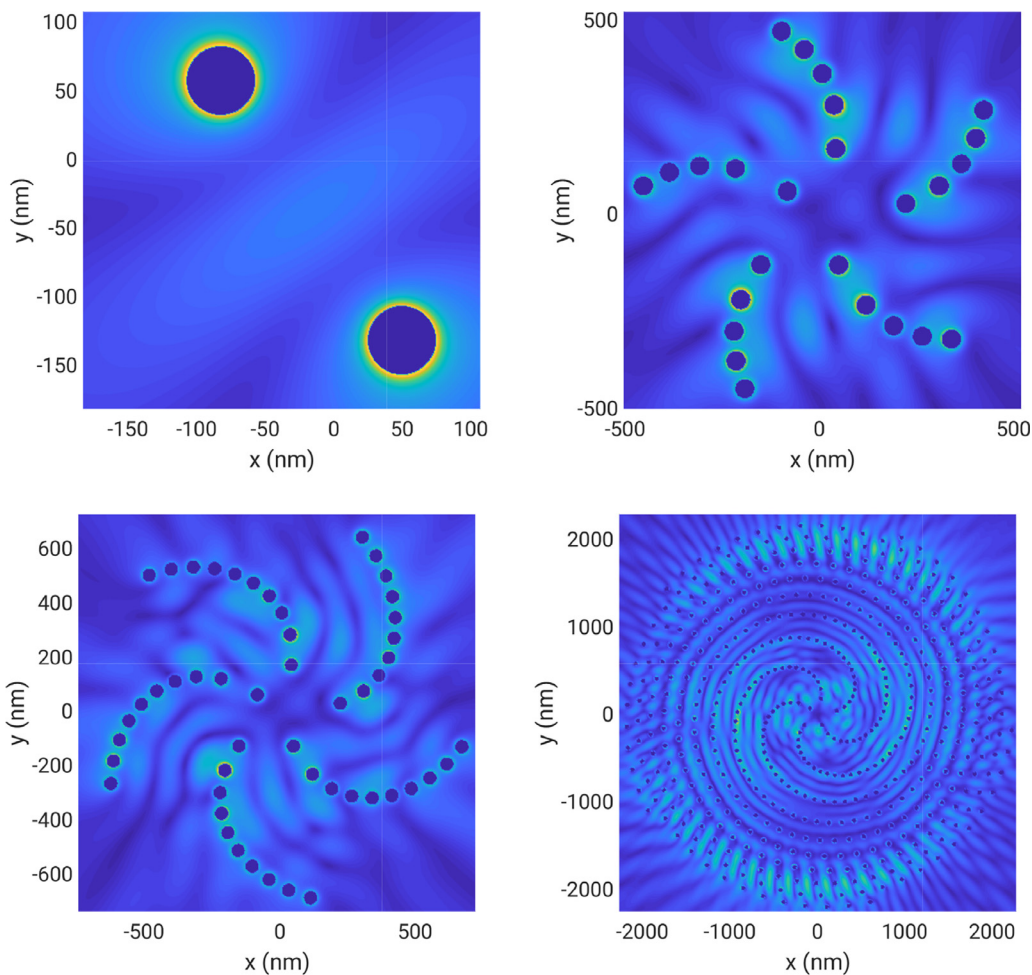
Fig. 13. Sample configurations of Vogel spirals with {2, 25, 50, 500} particles.
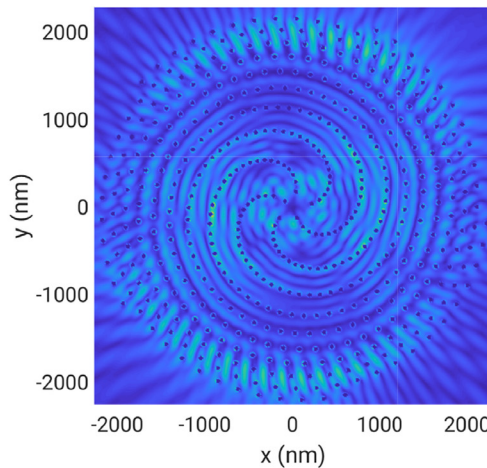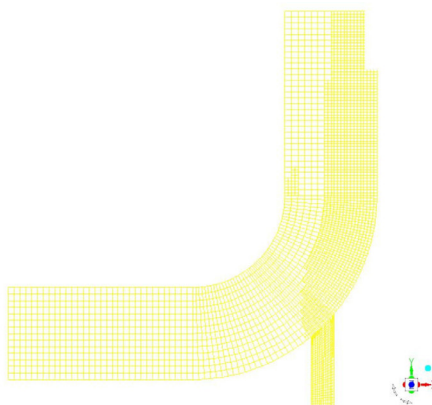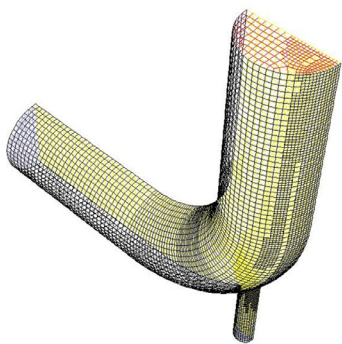

Fig. 14. Computational domain and geometrical configuration of an elbow pipe and its plane of symmetry; Test Problem 3.

For this problem, the high-dimensional output (dimension equal to number of grid points D) prohibits the direct implementation of Greedy-NAR and standard NAR [13,33] due to the use of an ARD kernel. As is shown in the decomposition approach in [33], the high dimensional outputs can be approximated using a linear combination of basis vectors in a reduced dimensional subspace of $\mathbb{R}^D$, of dimension $R \ll D$, i.e.,

$$\mathbf{y}^{(f)}(\boldsymbol{\xi}) = \sum_{r=1}^{R} z_r^{(f)}(\boldsymbol{\xi})\mathbf{v}_r^{(f)} = \mathbf{V}^{(f)}\mathbf{z}^{(f)}(\boldsymbol{\xi}), \tag{24}$$

where $\{\mathbf{v}_r\}_{i=1}^{D}$ form a basis for $\mathbb{R}^D$ and $z_r^{(f)}$ are the corresponding coefficients in this basis. R is chosen such that it captures a prescribed total variation in the data.

In [33], the bases are found using a proposed residual principal component analysis (resPCA) that enforces a uncorrelated structure, which contradicts our assumption that the F1 and F2 correlations are similar. Instead, we construct a universal basis by combining information from different fidelities. We define the basis as the eigenvector solutions of the following problem

$$\lambda \mathbf{v}_r = \mathbf{v}_r \bar{\mathbf{C}} \tag{25}$$

where $\bar{\mathbf{C}}$ is the covariance matrix of the average target function $\bar{\mathbf{y}}(\boldsymbol{\xi}) = \mathbb{E}[\mathbf{y}(\boldsymbol{\xi})] \approx (1/N_f) \sum_{n=1}^{N_f} \mathbf{y}^{(f)}(\boldsymbol{\xi}_n)$. Since the basis vectors are orthogonal, the representations are easily computed via

$$\mathbf{z}_n^{(f)} = \mathbf{y}_n^{(f)}\left(\mathbf{V}^{(f)}\right)^T. \tag{26}$$

For our problem, $R = 5$ basis vectors were sufficient to capture 99.5% of the total variance in the data. With the approximation (24), we can now treat $\mathbf{z}^{(f)}(\boldsymbol{\xi})$ as our multi-fidelity target functions and apply NAR, Greedy-NAR, and SC.

References

[1] Alvarez MA, Rosasco L, Lawrence ND. Kernels for vector-valued functions: a review. Found Trends Mach Learn 2012;4(3):195–266.

[2] Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, et al. Global sensitivity analysis: the primer. John Wiley & Sons, Ltd; 2008.

[3] Christofi AC, Pinheiro FA, Negro LD. Probing scattering resonances of Vogel's spirals with the Green's matrix spectral method. Opt Lett 2016;41(9):1933–6.

[4] Conti S, O'Hagan A. Bayesian emulation of complex multi-output and dynamic computer models. J Stat Plan Inference 2010;140(3):640–51.

[5] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., & González, J. (2019). Deep gaussian processes for multi-fidelity modeling. arXiv:1903.07320.

[6] Damianou A, Lawrence N. Deep Gaussian processes. In: Artificial intelligence and statistics; 2013. p. 207–15.

[7] Girard A, Rasmussen CE, Candela JQ, Murray-Smith R. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In: Advances in neural information processing systems; 2003. p. 545–52.

[8] Gratiet LL. Multi-fidelity gaussian process regression for computer experiments, Ph.D. thesis. Université Paris-Diderot-Paris VII; 2013.

[9] Guérin C-A, Mallet P, Sentenac A. Effective-medium theory for finite-size aggregates. J Opt Soc Am A 2006;23(2):349–58.

[10] Gunzburger MD, Peterson JS, Shadid JN. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. Comput Methods Appl Mech Eng 2007;196(4-6):1030–47.

[11] Hampton, J., Fairbanks, H., Narayan, A., & Doostan, A. (2017). Parametric/stochastic model reduction: low-rank representation, non-intrusive bi-fidelity approximation, and convergence analysis. arXiv:1709.03661.

[12] Hampton J, Fairbanks HR, Narayan A, Doostan A. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. J Comput Phys 2018;368:315–32.

[13] Higdon D, Gattiker J, Williams B, Rightley M. Computer model calibration using high-dimensional output. J Am Stat Assoc 2008;103(482):570–83.

[14] Jofre, L., Geraci, G., Fairbanks, H., Doostan, A., & Iaccarino, G. (2018). Multi-fidelity uncertainty quantification of irradiated particle-laden turbulence. arXiv:1801.06062.

[15] Kennedy MC, O'Hagan A. Predicting the output from a complex computer code when fast approximations are available. Biometrika 2000;87(1):1–13.

[16] Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J Royal Stat Soc: Ser B (Stat Methodol) 2001;63(3):425–64.

[17] Knezevic DJ, Nguyen N-C, Patera AT. Reduced basis approximation and a posteriori error estimation for the parametrized unsteady boussinesq equations. Math Models Methods Appl Sci 2011;21(07):1415–42.

[18] Krekeler C, Agarwal A, Junghans C, Praprotnik M, Delle L, Site L. Adaptive resolution molecular dynamics technique: down to the essential. J Chem Phys 2018;149(2):024104.

[19] Lee JG. Computational materials science: an introduction. CRC Press; 2016.

[20] Narayan A, Gittelson C, Xiu D. A stochastic collocation algorithm with multifidelity models. SIAM J Sci Comput 2014;36(2):A495–521.

[21] Perdikaris P, Raissi M, Damianou A, Lawrence N, Karniadakis GE. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, proceedings of the royal society a: Mathematical. Phys Eng Sci 2017;473(2198):20160751.

[22] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. MIT Press; 2006.

[23] Razi M, Kirby RM, Narayan A. Fast predictive multi-fidelity prediction with models of quantized fidelity levels. J Comput Phys 2019;376:992–1008.

[24] Razi M, Narayan A, Kirby RM, Bedrov D. Fast predictive models based on multi-fidelity sampling of properties in molecular dynamics simulations. Comput Mater Sci 2018a;152(C):125–33.

[25] Razi M, Wang R, He Y, Kirby RM, Negro LD. Optimization of large-scale Vogel spiral arrays of plasmonic nanoparticles. Plasmonics 2018b;14:253–61.

[26] Santner T, Williams B, Notz W. The design and analysis of computer experiments. Springer; 2003.

[27] Scholkopf B, Smola A, Muller K R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 1998;10(5):1299–319.

[28] Shah A, Xing W, Triantafyllidis V. Reduced-order modelling of parameter-dependent, linear and nonlinear dynamic partial differential equation models. Proc Royal Soc A: Math Phys Eng Sci 2017;473(2200):20160809.

[29] Shah AA, Luo K, Ralph T, Walsh FC. Recent trends and developments in polymer electrolyte membrane fuel cell modelling. Electrochim Acta 2011;56(11):3731–57.

[30] Skinner R, Doostan A, Peters E, Evans J, Jansen KE. An evaluation of bi-fidelity modeling efficiency on a general family of NACA airfoils. In: Proceedings of the 35th AIAA applied aerodynamics conference; 2017. p. 3260.

[31] Sobol' IM. On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 1967;7(4):784–802.

[32] Weber AZ, Mench MM, Meyers JP, Ross PN, Gostick JT, Liu Q. Redox flow batteries: a review. J Appl Electrochem 2011;41:1137–64.

[33] Xing, W., Kirby, R. M., & Zhe, S. (2019). Deep coregionalization for the emulation of spatial-temporal fields. arXiv:1910.07577.

[34] Xing W, Shah AA, Nair PB. Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on isomap. In: Proceedings of the royal society of london a: mathematical, physical and engineering sciences, vol. 471. The Royal Society; 2015. 2014–0697

[35] Zhu X, Narayan A, Xiu D. Computational aspects of stochastic collocation with multifidelity models. SIAM/ASA J Uncertain Quantif 2014;2(1):444–63.