INTERNATIONAL JOURNAL OF HYDROGEN ENERGY 48 (2023) 23242–23257

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/he

Multi-fidelity design optimization of solid oxide fuel cells using a Bayesian feature enhanced stochastic collocation

Wei W. Xing [b], Akeel A. Shah [a,*], Guohao Dai [c], Ziyang Zhang [d], Ting Guo [c], Hong Qiu [c], Puiki Leung [a], Qian Xu [e], Xun Zhu [a], Qiang Liao [a]

[a] Key Laboratory of Low-grade Energy Utilization Technologies and Systems, Chongqing University, Ministry of Education, Chongqing 400044, PR China

[b] School of Integrated Circuit Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing, 100191, China

[c] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, 518060, China

[d] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China

[e] Institute for Energy Research, Jiangsu University, Zhenjiang 212013, China

ARTICLE INFO

Article history:
Received 30 June 2022
Received in revised form
30 March 2023
Accepted 1 April 2023
Available online 29 April 2023

Keywords:
Solid oxide fuel cell optimization
Spatially distributed
Multi-fidelity
Stochastic collocation
Feature engineering
Bayesian

ABSTRACT

We develop a multi-fidelity surrogate modelling approach to replace the complex and costly physics-based computer models that are often used in the optimization of solid oxide fuel cell (SOFC) performance, or the simplified models that are used in lieu of complex models. We extend multi-fidelity stochastic collocation through a feature engineering step, and eliminate the requirement for the exact low-fidelity output at the inference stage. In contrast to previous approaches, the surrogate model we develop provides detailed spatial information, rather than one or more scalar outputs. This allows for the incorporation of such information into the objective of the optimization study, with the flexibility to choose from more than one objective, such as a minimum, maximum or average. Furthermore, the detailed spatial information can be used for general design purposes, such as ensuring uniformity in reactant and potential distributions. From the results on a 3-d SOFC model, we demonstrate highly accurate predictions of multiple spatially distributed quantities at up to spatial 250,000 locations. The results are superior to state-of-the-art multi-fidelity approaches, particularly for low numbers of high fidelity training points. We use the surrogate model to optimize the SOFC performance with respect to different objectives (including with nonlinear constraints and multiple objectives), with results that are accurate and are obtained in a fraction of the time required for the full model.

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

https://doi.org/10.1016/j.ijhydene.2023.04.012

Introduction

Fuel cells (FCs) are expected to play a major role in the future energy landscape, for standalone stationary power applications and for electric vehicle propulsion. FCs convert chemical energy into electricity [1,2] generating benign products. Solid oxide fuel cells (SOFC) are highly attractive due to their fuel flexibility, range of different geometries, and high temperatures, allowing for combined heat and power. The high temperatures lead to low costs since they eliminate the need for a catalyst and allow for fuels to be reformed internally in the anode. Modelling and simulation are important tools in the development of fuel cells, providing fundamental insights and lowering the costs associated with laboratory investigations, particularly for design and optimization [3,4].

Detailed fuel cell models, however, involve a complex system of nonlinear partial differential equations in a 3-d geometry comprising multiple components. They are often referred to as computational fluid dynamics (CFD) approaches, although in the general case they contain other laws such as charge and reactant conservation, and incorporate electrochemical reactions. Detailed reviews of SOFC and polymer electrolyte membrane (PEM) fuel cell modelling, including the various approaches and modelling assumptions, can be found in Refs. [5–7]. The time costs for solving such models, using finite differences, finite volumes or finite elements along with time-stepping schemes, can be prohibitive for applications that require many results at different parameter/input values, as for example, in optimization, sensitivity analyses and parameter identification. For small number of parameters [8] the time costs may be acceptable, but the number of computer model runs scales exponentially with the number of parameters, rendering complex models infeasible in many cases.

The most frequently employed strategy to circumvent this problem is to use simplified models. For example, Salva et al. [9] used a 1-d analytical model for optimizing PEM fuel cell performance with respect to operating conditions, while Hasanien et al. used a 0-d model for PEM fuel cell model parameter identification [10]. Pourkiaei et al. performed a multi-objective optimization of a tubular SOFC using a 0-d model combined with a genetic algorithm, using the electrical power and exergy destruction as objectives [11]. A similar study was performed by Mojaver et al. [12] for a methane-fueled SOFC, using the electrical energy efficiency, CO_2 emissions and electrical exergy efficiency as the objectives. A combined Taguchi/AHP/TOPSIS method was used for the optimization. A review of SOFC model parameter identification using similar models can be found in Ref. [13], based on a variety of optimization techniques.

Such approaches, however, will inevitably introduce crude simplifications that can lead to unsatisfactory results. Guo et al. used a detailed 3-d SOFC model that included charge, mass, momentum and heat balances. The model was developed in Ref. [14] and coupled to a mechanical-failure model in Ref. [15] to study the impact of the interconnector (IC) structure on the electrical performance and mechanical stability. Although the authors stated that their study was an optimization, in common with many other such studies, in actual fact it was a parametric study. Abdullah and Liu [16] performed a microstructural optimization for a low-temperature SOFC by coupling macroscopic and pore-scale models. To make the optimization feasible, the macroscopic model was restricted to 1-d.

An alternative approach is to use surrogate model approximations, usually based on machine learning or multi-fidelity methods. In the context of electrochemical energy technologies, machine learning methods have been used extensively as surrogate models for sensitivity analyses [17], optimization [18,19], fuel cell degradation [20], monitoring fuel cell performance [21], battery health monitoring [22,23], inverse parameter estimation [24] and model identification [25]. Examples of the methods employed are linear regression [26], artificial neural networks (ANNs) [27–29], deep learning networks (DNNs) [30,31] and Gaussian process (GP) models [21,32]. ANNs [28,29] and DNNs [33] have proven to be particularly popular given the availability of open-source and commercial codes.

In [34], Xu et al. assessed 19 different machine learning approaches, including ANNs, Boltzmann machines and support vector regression trained on data from a CFD model, which were used in combination with a genetic algorithm to optimize the power density output of a SOFC. Wang et al. [35] developed a response surface model based on linear regression using data from a finite element model to optimize geometric parameters for minimizing the long-term creep of a SOFC. Subotic et al. instead conducted an SOFC performance optimization using experimental polarization and impedance data, used for an ANN (although it was actual a parametric study) [29]. In general, however, ANNs require large volumes of data, which can be a hindrance when the complex computer model used to generate the data requires hours or even days to complete one simulation.

Almost exclusively, as in the examples cited above, machine-learning surrogate models have been used for scalar outputs, such as a peak power density or an efficiency. Ideally, however, the output of the surrogate model should contain the richness of information (spatial distributions of reactant concentrations, ionic and electronic potentials, temperature, and so on) found in the original computer model, in order to make fully-informed design and optimization decisions. For example, it is important to ensure uniform distributions of the reactants, temperature, flow velocity and overpotentials in the electrodes, which optimizes the performance and efficiency, and mitigates against degradation. This cannot be achieved by relying on a single or multiple scalar summaries from the model. Spatio-temporal information can also allow for optimization with respect to multiple objectives, and these objectives do not necessarily need to be decided beforehand. For example, we may use a minimum, maximum, average or some other functional summary. In this paper we therefore focus on the approximation of spatial outputs. Surrogate models for spatial or spatio-temporal outputs are particularly challenging due to the sizes of the output spaces. It is no surprise therefore that there have been few attempts to develop such models for fuel cells (and batteries), with exceptions being the reduced-dimensional GP approach of Shah and co-workers [36,37], and a similar approach combined with a coregionalisation model for multi-task learning (multiple outputs learned together) [38].

Another paradigm for developing surrogates is multi-fidelity modelling, in which data from models of low fidelity

(low accuracy and computational cost) is leveraged to aid the learning of outputs at a higher fidelity [39]. The great advantage of such approaches is that they typically require fewer runs of the high-fidelity model than are required for pure machine learning approaches, which rely entirely on high-fidelity data. When the number of system parameters (those of interest in the application) is high, collecting the data for training a pure machine learning model can itself become an issue.

In [40], Liu et al. developed a dual-segmented approach for a PEM fuel cell stack in which interpolation was used to bridge the gap between coarse- and fine-level segmentations. The authors claimed that the high fidelity result was equivalent to a finite-element simulation in terms of accuracy, but presented no direct evidence of this. Voskuilen et al. [41] surveyed various battery models, classifying them in terms of the fidelity, which was typically based on the number and extent of simplifications (of the underlying physics). The authors did not, however, combine these models in any way, in spite of the title of their paper. More conventional and sophisticated multi-fidelity approaches have not been applied to fuel cells. Xing et al. [42] developed a residual Gaussian process model that was tested in only one example on an SOFC finite-element model.

Many multi-fidelity models involve building a surrogate model (usually based on machine learning) at each fidelity. The linear autoregressive (LAR) multi-fidelity model [43] (see also [44]) assumes a linear relationship or parametric mapping between the different fidelity levels. Nonlinear AR (NARGP) [45,46] places GP priors over the unknown mappings between fidelities, allowing for greater model flexibility. For spatial and spatio-temporal problems, the major shortcoming of this method is that the solution at a fidelity f is used as an input to the GP model for fidelity $f + 1$. This presents an enormous challenge in terms of learning the model hyperparameters. Furthermore, tractability is lost and expensive sampling or variational methods are required for training and inference.

In multi-fidelity stochastic collocation (SC) [47], a greedy algorithm is used to identify inputs for a low number of high-fidelity simulations, while the low-fidelity results are used to approximate the coefficients in a high-fidelity SC approximation. SC and related approaches require out-of-sample low-fidelity model results in order to make predictions, which may be justified in some design contexts but renders them unusable for optimization, sensitivity analysis and uncertainty quantification when the low fidelity model is itself moderately expensive, as is frequently the case.

Greedy NAR [48] provides a bridge between NARGP and SC, appealing to the advantages of both methods. The high-fidelity model is written as a linear map of the low-fidelity model in an abstract (feature) space, with the feature map specified implicitly by marginalising over the weight matrix appearing in the map and using kernel substitution. Active learning is then used to efficiently identify the inputs for the data generation at all fidelities greater than the lowest. Greedy NAR outperformed LAR, NARGP and SC, but was subsequently outperformed by ResGP [42], which uses GP models for the residuals between fidelities, leading to a tractable posterior and scalable hyperparameter learning.

In this paper we develop a multi-fidelity surrogate model for a SOFC that is capable of approximating detailed spatial information and can be used for optimization purposes. We extend the SC approach to make it more accurate and apply it to a detailed 3-d SOFC model that includes mass, momentum and charge balances in the channels, electrodes and electrolyte. The model capacity of SC is enhanced by introducing a novel feature engineering step, and the requirement of low-fidelity data for predictions is eliminated by using a GP model. The approach is shown to yield highly accurate predictions of multiple spatially distributed quantities, including the electrolyte current density, ionic potential, volumetric anode current density and cathode overpotential, at up to 250,000 spatial locations. We note that these quantities are chosen for the purposes of illustration, and that there is no restriction on the type of field output, which can be related to the electrochemical performance, flow field, reactant concentrations or temperature. Indeed, SC has been successfully applied to frequency-modulated heat driven cavity flows [49], acoustics [50], molecular dynamics [51], airfoil design [52], discrete-space probability density evolution [53], and turbulence modelling [54].

The results are compared to LAR, NARGP, ResGP and Greedy NAR, demonstrating superior performance, particularly for low numbers of high fidelity training points. We also compare the computational complexities of these methods with that of ours. We then apply the surrogate model to the optimization of the SOFC performance, considering different objectives, objectives with nonlinear constraints and multiple objectives. The level of detail furnished by the surrogate model we develop allows for flexibility in deciding the objectives a-posteriori, in contrast to approaches that focus on scalar targets. We reiterate the novel aspects of our work: (i) as far as we are aware, we develop the first genuine multi-fidelity approach for SOFC that requires only a few runs of a complex (time-intensive), physics-based model to generate data; (ii) we are able to predict spatial outputs, allowing for more informed design and optimization decisions, with some flexibility to change objectives a-posteriori; (c) the method we develop significantly extends the original stochastic collocation formulation using feature enhancement and, moreover, replaces the expensive low-fidelity simulation required for prediction with an extremely rapid GP approximation.

Physics-based high and low fidelity models of a SOFC

We consider a detailed steady-state, isothermal 3-d solid oxide fuel cell model, which was implemented in the COMSOL Multiphysics software package[1]. We present only the essential details of the model and refer the reader to the documentation for the full set of equations and parameter values. A two-fidelity setting was considered. COMSOL Multiphysics uses the finite element method to solve the governing equations. Additionally, a V cycle geometric multigrid is used in this particular implementation. The domain comprises single anode and cathode channels, the electrodes and a ceramic separator, depicted in Fig. 1. The spatial variable is denoted $\mathbf{x} = (x,y,z)^T$. The dimensions of the channels are 1 cm $\times$ 0.5 mm $\times$

[1] https://www.comsol.com/model/current-density-distribution-in-a-solid-oxide-fuel-cell-514 (Application ID: 514).

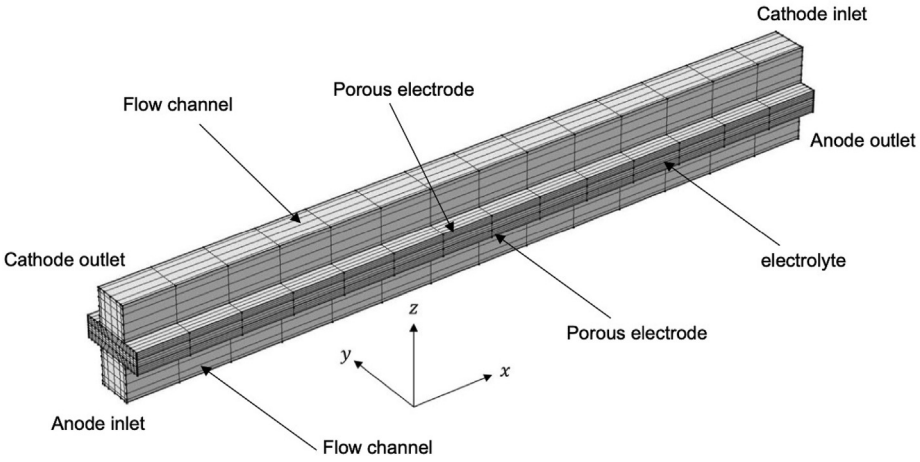

Fig. 1 – Computational domain for the COMSOL model.

0.5 mm in the x × y × z directions. The dimensions of the electrodes are 1 cm × 1 mm × 0.1 mm, and the dimensions of the separator are 1 cm × 1 mm × 0.1 mm. The geometry is constructed such that the anode inlet is placed at $x = 0$, while the cathode inlet is placed at $x = 1$ cm.

The model is based on hydrogen and air at the anode and cathode, respectively. Humidified hydrogen (hydrogen and water vapour) is supplied at the anode inlet, while humidified air (oxygen, water vapour and nitrogen) is supplied at the cathode inlet. The model includes charge balances in the electrolyte and electrodes (based on Ohm's law with fixed conductivity values) to calculate the electronic and ionic potentials. The incompressible Navier-Stokes equations along with continuity are used for the gas channel flow, while Brinkman's equation is used for the porous electrode flow. Mass balances of all species are solved in the channels and electrodes, incorporating Maxwell-Stefan diffusion and bulk motion by convection for the multi-component transport.

The electrochemical reactions are:

$$\begin{aligned} &H_2 + O^{2-} \rightarrow H_2O + 2e^- \quad \text{anode} \\ &O_2 + 4e^- \rightarrow 2O^{2-} \quad \text{cathode} \end{aligned} \tag{1}$$

Butler-Volmer laws are assumed for these reactions, with current densities (Am^{-2}) (and associated reaction rates via stoichiometry) given by:

$$\begin{aligned} i_a &= i_{0,a}\left(\frac{P_{H_2}}{P_{H_2,ref}}\exp\left(\frac{F\eta}{2RT}\right) - \frac{P_{H_2O}}{P_{H_2O,ref}}\exp\left(-\frac{3F\eta}{2RT}\right)\right) \quad \text{anode} \\ i_c &= i_{0,c}\left(\exp\left(\frac{3.5F\eta}{RT}\right) - \frac{P_{O_2}}{P_{O_2,ref}}\exp\left(-\frac{F\eta}{2RT}\right)\right) \quad \text{cathode} \end{aligned} \tag{2}$$

in which $i_{0,a}$ and $i_{0,c}$ are the anode and cathode exchange current densities (Am^{-2}),P_j and $P_{j,ref}$ denote the partial pressures and reference pressures of species j (Pa), F is Faraday's constant ($Cmol^{-1}$), R is the universal gas constant ($Jmol^{-1}K^{-1}$), T is the cell temperature (assumed constant) and η is the overpotential (V). The latter is defined by:

$$\eta = \phi_s - \phi_e - E_{eq,a} \tag{3}$$

in which ϕ_s and ϕ_e are the solid phase and electrolyte potentials, respectively, and E_{eq} is the equilibrium potential (assumed constant) for the anode or cathode ($j \in \{a, c\}$).

The cell is assumed to operate in potentiostatic mode. At anode channel rib, where the electrode interfaces with the end plate, a reference potential of zero is imposed, while the potential is set to the cell voltage E_{cell} at the cathode channel rib, where again the electrode interfaces with the end plate. At the walls of the gas channels and the porous electrode, insulation (zero mass and charge fluxes) is assumed. The compositions (weight fractions of species) are specified at the inlets, while the outlet conditions assume a convective flux. The default parameter values are specified in the documentation and are not repeated here to conserve space. Aside from the chosen inputs (specified below) all parameters were set to their default values.

Various settings can be adjusted in order to generate solutions of different fidelity. The low-fidelity solution in this case was defined by 360 domain elements, 474 boundary elements, 212 edge elements (corresponding to Fig. 1) and a relative tolerance of 0.1, while the high-fidelity solution was defined by 883,601 domain elements, 93,067 boundary elements, 3979 edge elements and a relative tolerance of 0.001, with other settings fixed, making the latter far more accurate and costly. The low fidelity model took 3 min and 3 s to compute the 128 solutions, while the high-fidelity model took 14 h, 17 min and 8 s on a Macbook pro 2.3 GHz, 8-Core Intel Core i9 with 64 GB 2667 MHz DDR4 RAM.

Four inputs were chosen for the experiments:

1. Electrode porosities in the range $\varepsilon \in [0.4, 0.85]$
2. Cell voltage in the range $E_{cell} \in [0.3, 0.95]$ V
3. Cell temperature (assumed constant) in the range $T \in [973, 1273]$ K
4. Channel pressures (assumed equal) in the range $P \in [0.5, 2.5]$ atm.

and are collected inside an input vector:

$$\xi = (\varepsilon, E_{cell}, T, P)^T \tag{4}$$

The ranges above define the feasible input space $\mathcal{D} \subset \mathbb{R}^p$, $p = 4$.

$N = 128$ values of each input (vector) were selected using a Sobol sequence design-of-experiment, and are labelled $\xi_n \in \mathcal{D}$,

$n = 1 \ldots, N$. The chosen outputs were: (1) the electrolyte current density $i_e(\mathbf{x}; \xi)$ (A m^{-2}); and (2) the ionic (electrolyte) potential $\phi_e(\mathbf{x}; \xi)$ (V), both in the $x - z$ plane at the centres of the channels in the y direction (see Fig. 1). This notation indicates that these outputs are functions of $\mathbf{x}$, and are parameterised by the inputs ξ.

Each of the outputs was recorded at locations $\mathbf{x}_j$, $j = 1, \ldots, d = 5000$ on a uniform grid, and were subsequently vectorised to form vectors $\mathbf{y}_n^f \in \mathbb{R}^d$, where the subscript n indicates that the input ξ_n was used, and f denotes the fidelity, namely $f = F$ for the high fidelity output, and $f = L$ for the low fidelity output. Note that we use a generic symbol $\mathbf{y}_n^f$ to denote both outputs in order to keep notation to a minimum.

Later in the optimization section Surrogate assisted optimization of an SOFC we also consider the volumetric current density in the anode $i_t(\mathbf{x}; \xi)$ (Am^{-3}) and the cathode overpotential $\eta_c(\mathbf{x}; \xi)$ (V), both recorded at 250,000 locations. We may think of the outputs as values of mappings (functions) $\mathbf{y}^f(\xi)$ of the input; that is, $\mathbf{y}_n^f = \mathbf{y}^f(\xi_n)$. Our ultimate goal is to find an estimator for the high-fidelity mapping $\mathbf{y}^F(\xi)$ so that rapid and accurate predictions can be made for an arbitrary ξ. This estimator can then be used for optimization, or any other application requiring repeated evaluations of, and predictions from the original computer model, e.g., sensitivity analysis and uncertainty quantification. The method we employ is an adaptation of the stochastic collocation (SC) approach [47], overcoming two of its limitations.

Stochastic collocation multi-fidelity fusion

The multi-fidelity SC methodology is based on a combination of sampling refinement and multi-fidelity data fusion. It has been successfully applied to numerous problems in science and engineering [50,52,54,55]. Consider low- and high-fidelity models that define mappings $\mathbf{y}^f : \mathcal{D} \to \mathbb{R}^d$ from a p-dimensional input parameter space $\mathcal{D}$ to low and high-fidelity output spaces lying in $\mathbb{R}^d$. In essence, the SC approach [47] finds an estimator $\mathbf{y}^F(\xi)$ of the high-fidelity mapping in the form:

$$\mathbf{y}^H(\xi) = \sum_{n=1}^{M} c_n(\xi)\mathbf{y}_n^H \tag{5}$$

in which M is the number of training points and $c_n(\xi)$ are unknown coefficients that depend on the input ξ. Clearly, this approach amounts to an interpolation, in which the coefficients (usually assumed to polynomials) are estimated using values of $\mathbf{y}^H(\xi)$ at judiciously selected collocation points, such as the Gauss-Lobatto-Legendre or Fekete points [56]. Due to the cost associated with calculating values of $\mathbf{y}^H(\xi)$, however, the low-fidelity solutions are instead employed to approximate the $c_n(\xi)$. Given $\mathbf{y}^L(\xi)$, corresponding to a general ξ, the coefficients are determined by the conditions:

$$\sum_{n=1}^{M} c_n(\xi)(\mathbf{y}_n^L)^T\mathbf{y}_n^L = (\mathbf{y}^L(\xi))^T\mathbf{y}_n^L, \quad \forall \mathbf{y}_n^L \tag{6}$$

This is an interpolation or projection of $\mathbf{y}^L(\xi)$ onto the space $\text{span}\{\mathbf{y}_n^L\}$, i.e., all linear combinations of the low fidelity data points. In practice, the SC approach is equivalent to a Gaussian process model with a linear kernel (Bayesian linear regression), as demonstrated in Ref. [48].

Assume that we are able to sample at random from the parameter space $\mathcal{D}$ and obtain N low-fidelity model evaluations. An $N \times N$ Gramian matrix $\mathbf{G}^L$ can be constructed as follows:

$$[G^L]_{l,n} = k(\mathbf{y}^L(\xi_l), \mathbf{y}^L(\xi_n)), \quad l, n = 1, \ldots, N \tag{7}$$

where $k(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is any valid kernel function. In the original formulation [47], a linear kernel is used, i.e.:

$$k(\mathbf{y}^L(\xi_l), \mathbf{y}^L(\xi_n)) = (\mathbf{y}^L(\xi_l))^T\mathbf{y}^L(\xi_n) \tag{8}$$

as seen in Eq. (6). A sampling refinement procedure to select the collocation points is then employed, involving the application of an LU decomposition with complete pivoting on the low-fidelity Gramian matrix $\mathbf{G}^L$. An ordering of the low-fidelity data points based on their 'importance' (explained in the next section) is obtained via the calculation of a matrix $\mathbf{Q}$ from:

$$\mathbf{P}\mathbf{G}^L\mathbf{Q} = \mathbf{L}\mathbf{U} \tag{9}$$

where the matrices $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular factorization matrices, respectively, while $\mathbf{P}$ and $\mathbf{Q}$ are permutation matrices that reorder the rows and columns of the Gramian matrix, respectively. This procedure leads to the selection of $m \ll N$ sampling points with indices $n_1, \ldots, n_m$, for determining the $c_n(\xi)$ and evaluating the high-fidelity output at any $\xi \in \mathcal{D}$. The latter is given by Eq. (5) applied at the sampling points:

$$\mathbf{y}^H(\xi) = \sum_{i=1}^{m} c_i(\xi)\mathbf{y}^H(\xi_{n_i}) \tag{10}$$

in which the $c_i(\xi)$ are computed via a least-squares projection onto the low-fidelity model data:

$$\begin{pmatrix} G^L_{i_1,i_1} & \cdots & G^L_{i_1,i_m} \\ \vdots & \ddots & \vdots \\ G^L_{i_1,i_m} & \cdots & G^L_{i_m,i_m} \end{pmatrix} \begin{pmatrix} c_1(\xi) \\ \vdots \\ c_m(\xi) \end{pmatrix} = \begin{pmatrix} k(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi_{i_1})) \\ \vdots \\ k(\mathbf{y}^L(\xi), \mathbf{y}^L(\xi_{i_m})) \end{pmatrix} \tag{11}$$

The $\mathbf{y}^H(\xi)$ in Eq. (10) is of course not the true latent function but an estimator. In order to avoid notational clutter, we do not make the distinction unless it is necessary. The same applies to the estimator of $\mathbf{y}^L(\xi)$ introduced later.

Explanation of the point selection strategy in multi-fidelity stochastic collocation

Selection of the inputs to generate the high-fidelity training data, i.e., the interpolation nodes ξ_{n_i} used in Eq. (10), is a weak greedy procedure. Consider a general description, in which we have a low-fidelity model $u^L(\xi) : \mathcal{D} \to \mathcal{V}^L$ in some low-fidelity solution space $\mathcal{V}^L$ that is assumed to be Hilbert space with inner product $\langle \cdot, \cdot \rangle^L$. Then the distance between any subspace $\mathcal{W} \subset \mathcal{V}^L$ and a function $v(\xi) \in \mathcal{V}^L$ can be defined as the greatest lower bound of the distance between v and $\mathcal{W}$, measured in terms of the norm induced by the inner product $\| \cdot \|^L = \langle \cdot, \cdot \rangle^L$:

$$d^L(v, \mathcal{W}) = \inf_{w \in \mathcal{W}} \|(I - P_{\mathcal{W}})v\|^L \tag{12}$$

where $P_{\mathcal{W}}$ denotes the orthogonal projection operator onto $\mathcal{W}$ and I is the identity operator. In our case, $u(\xi) = \mathbf{y}^L(\xi)$ and $\mathcal{V}^L \subset \mathbb{R}^d$, which is obviously a (finite dimensional) Hilbert space, for which we can use the standard inner product.

In radial basis methods (RBMs) for the (single-fidelity) solution $u(\xi) : \mathcal{D} \to \mathcal{V}$ to a parameter-dependent PDE problem, a common greedy strategy is to select R query points $\xi_1, \ldots, \xi_R$, to generate basis vectors $u(\xi_1), \ldots, u(\xi_R)$, from the functional manifold $\{u(\xi) : \xi \in \mathcal{D}\}$ such that the distance (as defined above) between $\{u(\xi) : \xi \in \mathcal{D}\}$ and $\text{span}\{u(\xi_1), \ldots, u(\xi_R)\} \subset \mathcal{V}$ is minimized.

We would like to perform a similar procedure to select query points for the high-fidelity data, but since the high-fidelity model is expensive, the low fidelity solution is instead used as a proxy. The selection of query points in RBMs as described above is implemented iteratively, by adding a point ξ^* to the optimal set Γ_n at step n such that the distance between uL(ξ^*) and $\mathcal{U}^L(\Gamma_n) = \text{span}\{u(\xi) | \xi \in \Gamma_n\}$ is maximal:

$$\xi_* = \underset{\xi \in \bar{\mathcal{D}}}{argmax} d^L(u^L(\xi), \mathcal{U}^L(\Gamma_n)), \quad \Gamma_{n+1} = \Gamma_n \cup \{\xi_*\} \tag{13}$$

in which $\overline{\mathcal{D}} \subseteq \mathcal{D}$ and $\Gamma_0 = \{\}$, i.e., the empty set. Performing this process in the continuous space $\overline{\mathcal{D}} = \mathcal{D}$ is generally not computationally feasible, due to the potentially very high number of queries to the low fidelity model u^L allied with the lack of a closed form expression for $\nabla_\xi u^L$, necessitating expensive numerical approximations. It is, however, achievable in a discrete finite-cardinality set $\overline{\mathcal{D}} \subset \mathcal{D}$.

As proven by Narayan et al. [47], full-pivoting LU decomposition of the low-fidelity Gramian matrix $\mathbf{G}^L$ to obtain the m indices $n_1, \ldots, n_m$, and corresponding sampling points $\Gamma_m = \{\xi_{n_i}\}$ is one of the greedy solutions to the optimization problem defined in Eq. (13) on the finite-cardinality candidate set $\overline{\mathcal{D}} = \{\xi_n\}_{n=1}^M$. This assumes that $\overline{\mathcal{D}}$ is dense enough to capture all important variations in the high-fidelity data, and that there are no restrictions on selecting point in $\mathcal{D}$.

Bayesian nonparametric model for the low-fidelity output when making predictions

In the original framework [47], a high-fidelity prediction $\mathbf{y}^H(\xi)$ for an unseen input ξ requires the true low-fidelity output $\mathbf{y}^L(\xi)$, which can still be very expensive to obtain in real-wold applications. To build a more efficient mutli-fidelity model, we use a GP approximation for the low-fidelity output by introducing a prior:

$$\mathbf{y}^L(\xi) \sim \mathcal{GP}(\mathbf{y}^L(\xi) | \mathbf{0}, k(\xi, \xi' | \boldsymbol{\theta}) \otimes \mathbf{Q} + \sigma^2 \mathbf{I} \otimes \delta(\xi, \xi')) \tag{14}$$

in which $\mathcal{GP}(\cdot | \cdot, \cdot)$ denotes a GP, with the first and second arguments denoting the mean and covariance functions. The symbol ~ signifies that the random variable or process to left is distributed according to the distribution to the right, $\otimes$ is the Kronecker product, $\delta(\xi, \xi')$ is the Kronecker-delta function and $\mathbf{Q}$ is a coregionalization matrix (treated as hyperparameter), containing the covariances between the components of $\mathbf{y}^L(\xi)$. $k(\xi, \xi' | \boldsymbol{\theta})$ is a kernel for the covariance across inputs, containing a vector of hyperparameter $\boldsymbol{\theta}$. The product structure $k(\xi, \xi' | \boldsymbol{\theta}) \otimes \mathbf{Q}$ is referred to as separable [57]. The term $\sigma^2 \mathbf{I} \otimes \delta(\xi, \xi')$ accounts for i.i.d. noise in the data, or equivalently acts as a regularization (penalty) term.

One of the most widely used kernels is the squared exponential automatic relevance determination (ARD) kernel:

$$k(\xi, \xi' | \boldsymbol{\theta}) = \exp\left(- \sum_{l=1}^{d} \frac{|\xi_{n,l} - \xi'_{n,l}|^2}{\theta_l^2} \right) \tag{15}$$

in which $\xi_{n,l}$ is the $l-$th component of ξ_n, and θ_l are correlation lengths that form the vector $\boldsymbol{\theta}$. The total set of hyperparameters is $\{\boldsymbol{\theta}, \mathbf{Q}, \sigma^2\}$.

The simulation data can be collected in a matrix $\mathbf{Y}^L = [\mathbf{y}_1^L, \ldots, \mathbf{y}_{N_L}^L]^T \in \mathbb{R}^{N_L \times d}$, where N_L is the number of low fidelity data points used for training. The log marginal likelihood $\mathcal{L}$ for the model 14 is:

$$\mathcal{L} = \frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \text{tr}\left(\text{vec}(\mathbf{Y}^L) \text{vec}(\mathbf{Y}^L)^T \boldsymbol{\Sigma}^{-1} \right) - \frac{dN_L}{2} \ln(2\pi) \tag{16}$$

in which vec($\cdot$) denotes a vectorization and tr is the trace of a matrix. $\boldsymbol{\Sigma} = \mathbf{Q} \otimes \mathbf{K} + \sigma^2 \mathbf{I} \in \mathbb{R}^{dN_L \times dN_L}$ is the covariance matrix, in which $\mathbf{K} \in \mathbb{R}^{N_L \times N_L}$ is given by $[K]_{ij} = k(\xi_i, \xi_j)$. A maximum likelihood estimate (MLE) can be used for the hyperparameters, which can be placed in the following posterior distribution for $\mathbf{y}^L(\xi)$ [58]:

$$\begin{aligned} \mathbb{E}[\mathbf{y}^L(\xi)] &= (\mathbf{Q} \otimes \mathbf{k}(\xi))^T \boldsymbol{\Sigma}^{-1} \text{vec}(\mathbf{Y}^L) \\ \text{Var}(\mathbf{y}^L(\xi)) &= k(\xi, \xi) \mathbf{Q} - (\mathbf{Q} \otimes \mathbf{k}(\xi))^T \boldsymbol{\Sigma}^{-1} (\mathbf{Q} \otimes \mathbf{k}(\xi)) \end{aligned} \tag{17}$$

in which $\mathbf{k}(\xi) = [k(\xi, \xi_1), \ldots, k(\xi, \xi_{N_H})]^T$ contains the correlations between the output at a test input ξ and the observations in the data set.

Efficient computation using a noise-free assumption

For multi-fidelity fusion problems, the noise term can, and in fact usually is, taken to be zero by virtue of the fact that the low- and high-fidelity models are deterministic [47]. With this assumption, the predictive mean in Eq. (17) simplifies as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{y}^L(\xi)] &= (\mathbf{Q} \otimes \mathbf{k}(\xi))^T (\mathbf{Q} \otimes \mathbf{K})^{-1} \text{vec}(\mathbf{Y}^L) \\ &= \mathbf{Q} \otimes (\mathbf{k}(\xi))^T (\mathbf{Q}^{-1} \otimes \mathbf{K}^{-1}) \text{vec}(\mathbf{Y}^L) \\ &= (\mathbf{Q}\mathbf{Q}^{-1}) \otimes (\mathbf{k}(\xi))^T \mathbf{K}^{-1} \text{vec}(\mathbf{Y}^L) \\ &= \mathbf{I} \otimes (\mathbf{k}(\xi))^T \mathbf{K}^{-1} \text{vec}(\mathbf{Y}^L) \\ &= \text{vec}\left((\mathbf{k}(\xi))^T \mathbf{K}^{-1} \mathbf{Y}^L \mathbf{I}^T \right) \\ &= (\mathbf{Y}^L)^T \mathbf{K}^{-1} \mathbf{k}(\xi) \end{aligned} \tag{18}$$

demonstrating that the value of $\mathbf{Q}$ does not affect the mean predictions. What is more, outputs in high-dimensional spaces (large d) can be handled without the need for the difficult task of learning the $d(d+1)/2$ components of $\mathbf{Q}$. To illustrate the difficulty in the optimization of (16), in the examples we consider, $d \in \{5000, 250{,}000\}$, yielding 1.25025×10^7 or 3.125×10^{10} entries. This would severely compromise the accuracy of the maximum likelihood estimate of the hyperparameters.

Therefore, we set $\mathbf{Q} = \mathbf{I}$ to obtain:

$$\mathbf{y}^L(\xi) \sim \mathcal{GP}\left(\mathbf{y}^L(\xi)|\mathbf{0}, k(\xi,\xi') \otimes \mathbf{I}\right) \tag{19}$$

and the likelihood:

$$\mathcal{L} = \frac{d}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left((\mathbf{Y}^L)^T\mathbf{K}^{-1}\mathbf{Y}^L\right) - \frac{N_L d}{2}\ln(2\pi) \tag{20}$$

The computational complexity associated with this likelihood is now $\mathcal{O}(N_L^3 d)$, in contrast to the $\mathcal{O}(N_L^3 d^3)$ complexity of the original likelihood maximization, making it significantly more scalable to high values of d. The corresponding predictive posterior is as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{y}^L(\xi)] &= \mathbf{k}(\xi)^T\mathbf{K}^{-1}\mathbf{Y}^L \\ \mathrm{Var}(\mathbf{y}^L(\xi)) &= \left(k(\xi,\xi) - \mathbf{k}(\xi)^T\mathbf{K}^{-1}\mathbf{k}(\xi)\right)\otimes\mathbf{I} \end{aligned} \tag{21}$$

which constitutes the estimator for the true low fidelity mapping.

Feature enhancement for the low-fidelity Gramian: an efficient nonlinear kernel

We note that the predictive equations in Eqs. (10) and (11) are very similar to those in Eq. (21). As we pointed out in Ref. [48], SC is a special case of NAR when using a linear kernel. However, in practice, we find that introducing a nonlinear kernel, e.g., an ARD kernel, can only improve the performance when sufficient data is provided to accurately learn the hyperparameters introduced; for ARD there are $\mathcal{O}(d)$ hyperparameters, which is typically very high for the types of models under consideration. In a practical situation in which the high-fidelity observations are sparse, performance will therefore be compromised.

There are possible remedies via approximate or fully Bayesian approaches, e.g., variational Bayes or MCMC, which, however, are computationally expensive and can defeat the original purpose of finding an inexpensive surrogate for the high fidelity model. To overcome this problem, we can use a feature enhancement technique to augment the low-fidelity observations in order to make more accurate high-fidelity predictions.

Specifically, we design a feature transformation of the low-fidelity observations via a linear combination of kernels as follows:

$$\begin{aligned} [G^L]_{l,j} &= \langle \mathbf{y}^L(\xi_l), \mathbf{y}^L(\xi_j)\rangle + \omega_1\langle (\mathbf{y}^L(\xi_l))^2, (\mathbf{y}^L(\xi_j))^2\rangle + \omega_2\langle \exp(\mathbf{y}^L(\xi_l)), \exp(\mathbf{y}^L(\xi_j))\rangle \\ &\quad + \omega_3\langle \ln(\mathbf{y}^L(\xi_l)), \ln(\mathbf{y}^L(\xi_j))\rangle + \omega_4\langle \sin(\mathbf{y}^L(\xi_l)), \sin(\mathbf{y}^L(\xi_j))\rangle + \ldots \end{aligned} \tag{22}$$

where $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product, ω_k are weights and the operations $(\cdot)^2$, $\exp(\cdot)$, etc. are performed element-wise. Other operations included in this work are $\sqrt{\cdot}$, $|\cdot|$ and $\cos(\cdot)$, although any arbitrary number Z of functions can be incorporated. In order to avoid overfitting, we place a Laplace prior over each weight:

$$p(\omega_k) \propto \exp(-|\omega_k|) \tag{23}$$

and set $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_Z)^T$. The Laplace prior with a maximum-likelihood point estimate of $\boldsymbol{\omega}$ is then equivalent to minimizing the L_2 loss function between the data points and their SC approximations with an L_1 regularization term:

$$\boldsymbol{\omega}^* = \underset{\omega}{argmin}\frac{1}{2}\sum_{n=1}^{N_H}\|\mathbf{y}^H(\xi_n) - \mathbf{y}_n^H\|^2 + \lambda\|\boldsymbol{\omega}\|_1 \tag{24}$$

in which: $\|\cdot\|_1$ is the L_1 norm; N_H is the number of high fidelity samples available ($N_H = m$ with the greedy strategy); $\mathbf{y}^H(\xi_n)$ are the SC predictions (functions of $\boldsymbol{\omega}$) via Eqs. (10), (11) and (22) of the high-fidelity observations $\mathbf{y}_n^H$; and λ is the regularization constant, which is determined using a leave-one-out cross-validation.

High-fidelity predictive posterior using Monte-Carlo integration

Given $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, we can derive the high-fidelity posterior distribution. Despite the fact that the low-fidelity prediction (21) for $\mathbf{y}^L(\xi)$ is Gaussian, due to the feature enhancement, the high-fidelity prediction $\mathbf{y}^H(\xi)$ is no longer tractable, which necessitates integration:

$$p(\mathbf{y}^H(\xi)) = \int p(\mathbf{y}^L(\xi))p(\mathbf{y}^H(\xi)|y^L(\xi))d\mathbf{y}^L(\xi) \tag{25}$$

in which $p(\mathbf{y}^L(\xi))$ is the predictive posterior in Eq. (21) and $p(\mathbf{y}^H(\xi)|y^L(\xi))$ is the nonlinear deterministic mapping given by Eqs. 10 and 11 and (22). If d is low, says less than 10, we can use an efficient Gaussian quadrature to approximate the integral. For observations in high-dimensional spaces, we can use a Monte-Carlo integration, which is more expensive but still efficient since sampling from a standard Gaussian distribution is straightforward and quick using reparameterization.

To this end, we first sample Kd points γ_i from a normal distribution $\mathcal{N}(0,1)$ and reshape them into a matrix $\boldsymbol{\Gamma}\in\mathbb{R}^{d\times K}$, with columns $\boldsymbol{\gamma}_k\in\mathbb{R}^d$, $k = 1, \ldots, K$. We can then approximate the mean of the high-fidelity predictive distribution as:

$$\mathbb{E}[\mathbf{y}^H(\xi)] \approx \sum_{k=1}^{K}(\mathbf{y}_k^H(\xi)|\mathbf{y}_k^L(\xi)) = \frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{m}c_l^k(\xi)\mathbf{y}^H(\xi_l) \tag{26}$$

in which

$$\mathbf{y}_k^L(\xi) = \mathbf{k}(\xi)^T\mathbf{K}^{-1}\mathbf{Y}^L + \gamma_k\sqrt{k(\xi,\xi) - \mathbf{k}(\xi)^T\mathbf{K}^{-1}\mathbf{k}(\xi)} \tag{27}$$

are samples from the low-fidelity predictive posterior (21). The coefficients $c_l^k(\xi)$ are obtained by substituting the low-fidelity samples into Eq. (11) with the kernel defined in Eq. (22). Similarly, the variance of the predictive distribution is obtained from:

$$\mathrm{Var}(\mathbf{y}^H(\boldsymbol{\xi})) = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{l=1}^{m} c_l^k(\boldsymbol{\xi})\mathbf{y}^H(\boldsymbol{\xi}_l) - \mathbb{E}[\mathbf{y}^H(\boldsymbol{\xi})]\right)\left(\sum_{l=1}^{m} c_l^k(\boldsymbol{\xi})\mathbf{y}^H(\boldsymbol{\xi}_l) - \mathbb{E}[\mathbf{y}^H(\boldsymbol{\xi})]\right)^T \tag{28}$$

Computational complexity

Table 1 compares the time/computational complexity for the training of our method with those of state-of-the-art methods. For a GP model with N_L training points, the computational complexity of the training is $\mathcal{O}(N_L^3)$, by virtue of the requirement to invert the $N_L \times N_L$ covariance matrix when implementing the maximum log-likelihood solution [58]. For SC, the complexity is $\mathcal{O}(N_H^3)$ due to the inversion of an $N_H \times N_H$ Grammian matrix in the loss function (24). This is the same complexity as that of all the other methods in Table 1, expect for linear autoregression (LAR) [43], which is considerably higher. The number of model parameters is comparable to ResGP [42] and LAR for typical values of p, but much lower than for NARGP [45] and Greedy NAR [48].

Results and discussion

In the comparisons in this section, NARGP, LAR, Greedy NAR and ResGP are implemented according to their original formulations and specifications. The GP based methods assume a zero-mean function after centering the data and ARD kernels are used in all of these methods, except for NARGP. For NARGP, the fidelity-1 kernel is an ARD kernel, and the fidelity 2 kernel is of the form used by Perdikaris et al. in their original implementation [45]. Three ARD kernels are combined together by factoring the dependence on the inputs and the low fidelity solution, and adding another kernel that represents a bias term. All methods were implemented in both Python (PyTorch) and Matlab. For the optimization, Matlab was preferred due to the availability of the Optimization Toolbox.

Table 1 – Model complexity comparison. p is the dimension of the input space, d is the dimension of the output space, N_H and N_L are the number of high and low fidelity training points, respectively, and Z is the number of features used in Eq. (22).

Method	Complexity	Number of parameters
Our method	$\mathcal{O}(N_L^3) + \mathcal{O}(N_H^3)$	$(p+1)+Z$
NARGP	$\mathcal{O}(N_L^3) + \mathcal{O}(N_H^3)$	$2(p+1)+d$
LAR	$\mathcal{O}((N_L+N_H)^3)$	$2(p+2)-1$
Greedy NAR	$\mathcal{O}(N_L^3) + \mathcal{O}(N_H^3)$	$(p+1)+d$
ResGP	$\mathcal{O}(N_L^3) + \mathcal{O}(N_H^3)$	$2(p+1)$

The prediction accuracy of all methods is measured using the root square mean error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_t}\sum_{n=1}^{N_t}\left\|\mathbf{y}_{p,n} - \mathbf{y}_n\right\|^2} \tag{29}$$

in which N_t is the number of test points, $\mathbf{y}_{p,n}$ is the prediction and $\mathbf{y}_n$ is the test point. All numerical experiments were conducted five times with randomization of the data, and the average value of the RMSE is presented. For individual results we use the square error (SE), $\|\mathbf{y}_{p,n} - \mathbf{y}_n\|^2$.

Before moving onto the multi-fidelity results, we first show (Fig. 2) box plots of the square errors (SE) against the test examples for the prediction of the low fidelity ionic potential $\phi_e(\mathbf{x};\boldsymbol{\xi})$ (V) and current density $i_e(\mathbf{x};\boldsymbol{\xi})$ (A m^{-2}) using Eq. (21), for an increasing number of training points. There is a clear decrease in the SE values as the number of training points increases, and the results are very accurate even at 40 training points.

To give and indication of the significance of the sizes of the errors, we first note that the mean value of the current density data was 1.058×10^3 A m^{-2}. The square root of the median square error per coordinate of the output ($d = 5000$ in this case) at 40 training points is 5.963 A m^{-2}, while the square root of the maximum square error (an outlier) is 50.178 A m^{-2}. The corresponding errors are significantly lower at 80 training points. For the ionic potential, the mean value of the data was -0.0924 V. The square root of the median square error per coordinate at 40 training points is 4.472×10^{-4} V, while the square root of the maximum square error (an outlier) is 1.890×10^{-3}. Again, the corresponding errors decrease significantly at 80 training points.

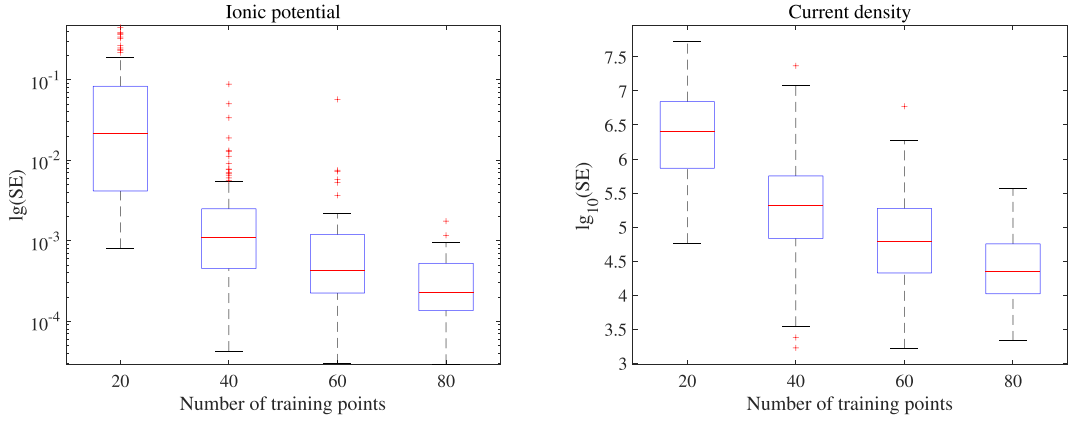

Fig. 2 – Box plots of the square errors (SE) against the test examples for the prediction of the low fidelity ionic potential $\phi_e(\mathbf{x};\boldsymbol{\xi})$ (V) and current density $i_e(\mathbf{x};\boldsymbol{\xi})$ (A m^{-2}) using Eq. (21), for an increasing number of training points.

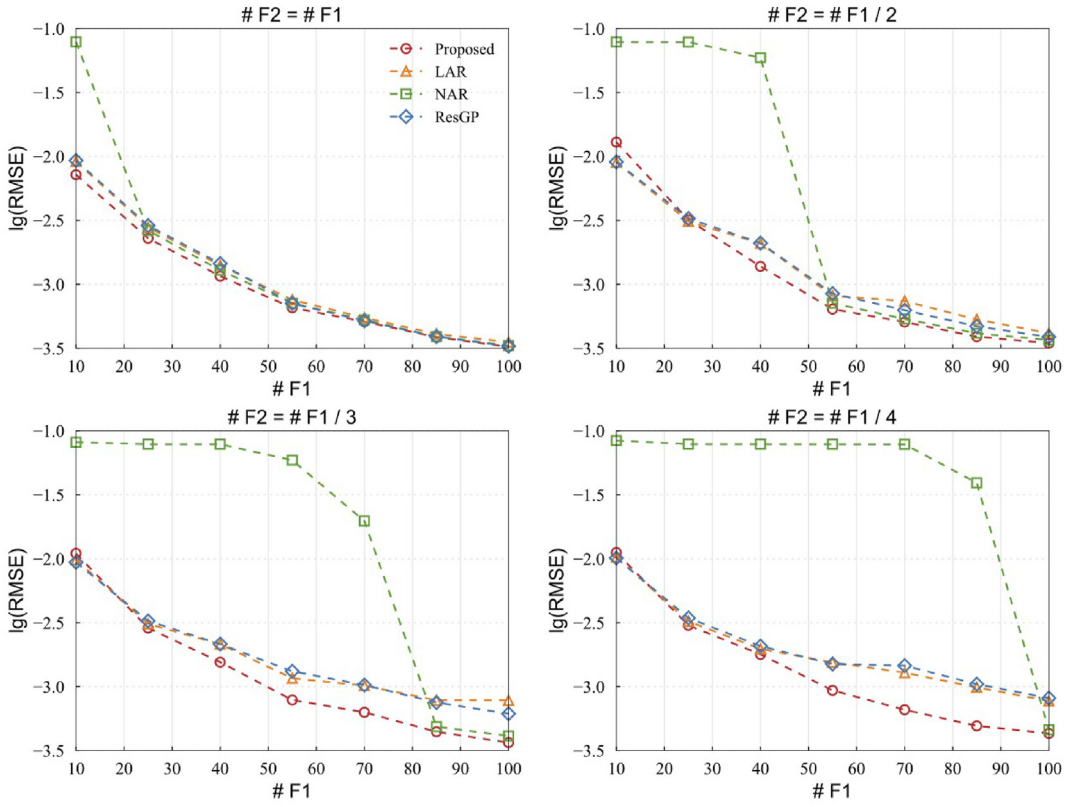

Fig. 3 – Log root mean square errors (RMSE) against the test examples for our proposed method, corresponding to different numbers and ratios of low- (F1) and high-fidelity (F2) ionic potential $\phi_e(\mathbf{x};\xi)$ (V) training data points. Included are comparisons to LAR, NARGP, ResGP, and Greedy NAR.

Prediction of the ionic potential

The RMSE values (with the five-fold cross validation) relating to the discretised ionic potential $\phi_e(\mathbf{x};\xi)$ (V) in the $x-z$ plane at the channel centres are shown in Fig. 3, for increasing numbers of training samples, with the high-to low-fidelity sample ratios fixed to 1:1, 1:2, 1:3, and 1:4. In each case the remaining high fidelity data points were used for testing. Comparisons to the other methods are also shown. When the numbers of low and high fidelity samples are equal, all methods yield a similar level of accuracy, with the exception of NARGP at low training point numbers. This is not surprising considering the number of hyperparameters involved in NARGP.

The accuracy of our method is very high, with average RMSE values below 1.65×10^{-5} V per each of the 5000 coordinates of the output when the number of low fidelity samples is more than around 25 (for all training data ratios). This is ca. 0.017% of the mean value -0.0970 V of the ionic potential across the data set. For around 55 training points, the RMSE is below 1×10^{-5} V per coordinate. As the ratio of high-fidelity to low-fidelity samples decreases, our method shows a significant improvement over the other methods, even at a low to mid-range number of training points (40–60). NARGP performs poorly except when a large data set is available; around 85 for a ratio of 3:1 and around 100 for a ratio of 1:4, in which case it approaches the accuracy of our method. As already explained, this is to be expected for high values of d.

The improvement over the other methods for decreasing high-fidelity to low-fidelity sample ratios is significant because it allows for fewer high-fidelity training points. For example, at a ratio of 4:1, with 72 low-fidelity and only 18 high-fidelity training points, our method is approximately 26% more accurate than LAR, the next best method. LAR, moreover, has roughly double the computational complexity compared to our method for this number of low and high fidelity samples.

We now compare our method to SC, in which both methods are provided with the exact low fidelity data for prediction. This allows us to assess the effect of the added features. The number of low and high fidelity training points is the same for both methods. As can be seen in Fig. 4, the methods are roughly the same for training point numbers below 25, but for higher numbers of training points our method is significantly more accurate. For training point numbers above 70, the RMSE is ca. 25% lower. Even at 55 training points it is more than 11% lower. This underlines the benefit of the added features.

We note also that in Fig. 4 the RMSE is significantly lower for both methods than the errors in Fig. 3. This is because the low fidelity data is provided for both methods, whereas in Fig. 3 the GP model (21) is used to predict the low-fidelity result in order to generate the high-fidelity approximation. Thus, it is not a fair comparison since execution of the low fidelity model involves considerable computational costs. For the optimization presented later (and other applications) it is usually infeasible to make repeated calls to the low fidelity model.

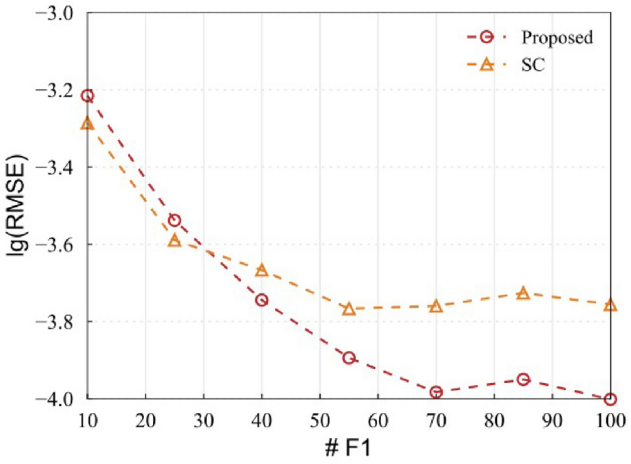

Fig. 4 – A comparison of log RMSE for stochastic collocation (SC) and our proposed method with the feature enhancement in Eq. (22). In this case, the exact low fidelity data is provided for prediction.

Examples of the predictions using our method are shown in Fig. 5, corresponding to the case with the lowest square error, a case in the region of the median square error and the case with the highest square error. Also shown are the tests (ground truths) and the point-wise absolute differences between the prediction and ground truth. 80 low-fidelity and 20 high-fidelity training points were used to generate these results. It can be seen that even in the case of the highest square error, the prediction is highly accurate, both qualitatively and quantitatively.

Prediction of the current density

The RMSE values (with the five-fold cross validation) relating to the discretised current density $i_e(\mathbf{x};\xi)$ (A m^{-2}) in the $x-z$ plane at the channel centres are shown in Fig. 6, for increasing numbers of training samples and different high-to low-fidelity sample ratios. In each case the remaining high fidelity data points were used for testing. Again, comparisons to the other methods are provided. As with the ionic potential, the RMSE

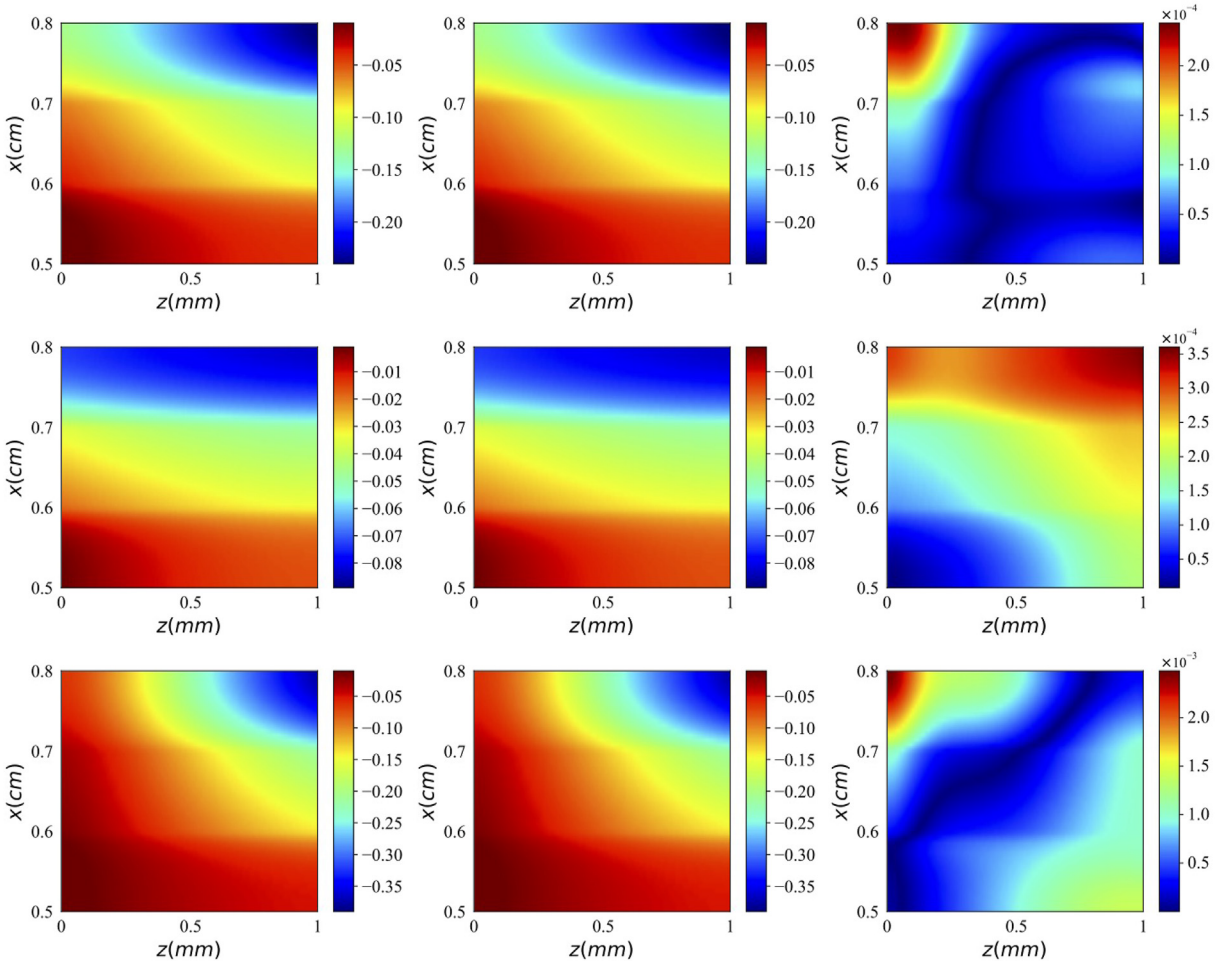

Fig. 5 – Three predictions of the ionic potential $\phi_e(\mathbf{x};\xi)$ (V) in the $x-z$ plane at the channel centres (Fig. 1) for 80 low-fidelity and 20 high-fidelity training points. These predictions were chosen such that the top row exhibited the lowest square error, the middle row close to the median square error and the bottom row the highest square error. The first column contains the prediction, the second column the tests (ground truths) and the third column the point-wise absolute differences.

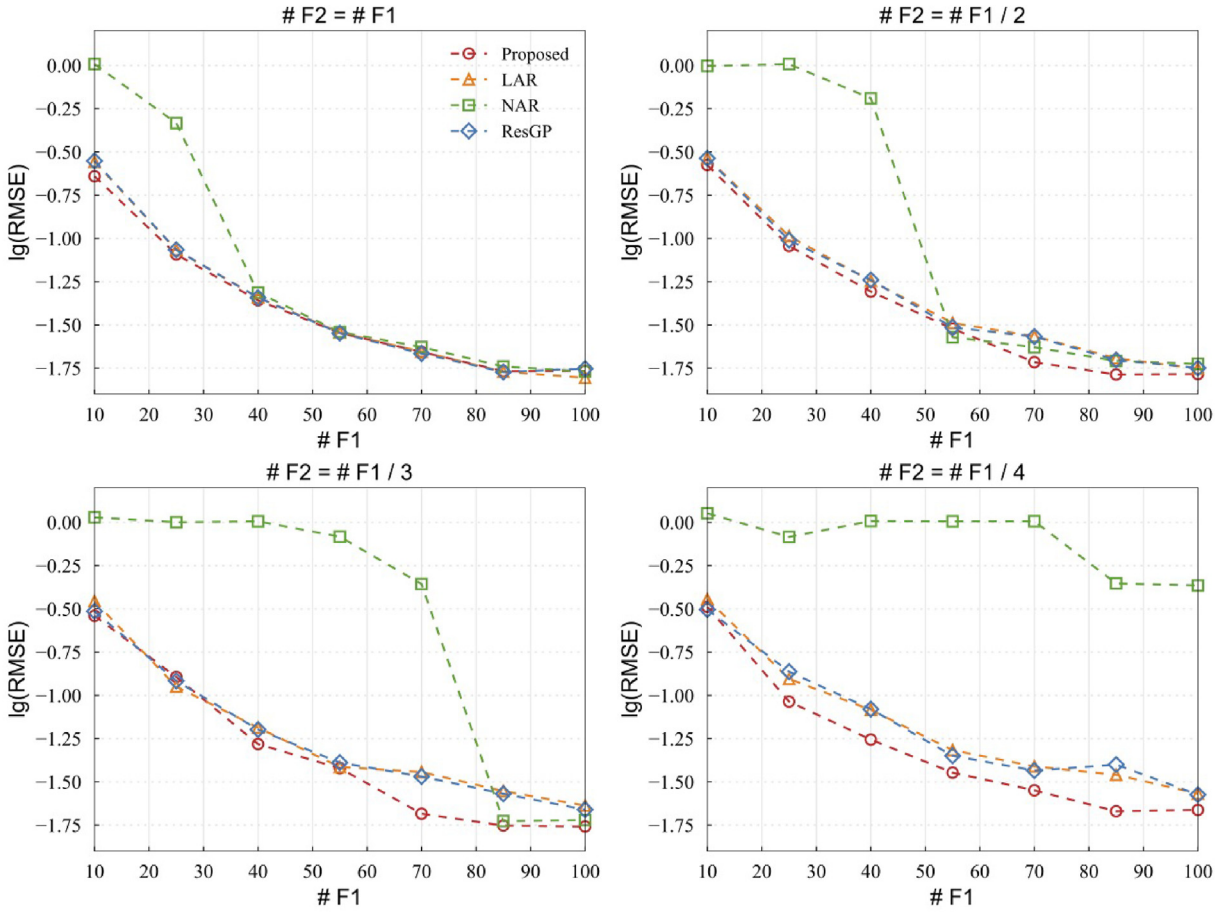

Fig. 6 – Log root mean square errors (RMSE) against the test examples for our proposed method, corresponding to different numbers and ratios of low- (F1) and high-fidelity (F2) current density $i_e(\mathbf{x};\xi)$ (A m^{-2}) training data points. Included are comparisons to LAR, NARGP, ResGP, and Greedy NAR.

values are roughly the same when the numbers of low and high-fidelity samples are equal. This, however, is unrealistic. The more realistic scenarios of 1:2, 1:3, and 1:4 show again that our method yields the lowest RMSE in most cases, and the differences are especially noticeable at the two lowest ratios. It must be said, however, that the differences are not as pronounced as in the case of the ionic potential. At a training ratio of 1:3, the RMSE is, nevertheless, ca. 20% lower than it is for ResGP, the next best method.

Examples of the predictions using our method are shown in Fig. 7, corresponding to the case with the lowest square error, a case with a square error in the vicinity of the median square error and the case with the highest square error, together with the ground truths and the point-wise absolute differences. 80 low-fidelity and 20 high-fidelity training points were used. Again, even in the case of the highest square error, the prediction is highly accurate.

Surrogate assisted optimization of an SOFC

The results of the previous section demonstrate that our method has sufficient accuracy to be used as surrogate model for optimization. In this section we employ the method to optimize the SOFC performance with respect to several objectives. We note that the objectives are illustrative and serve as examples and tests, rather than being physically inspired.

We start with the aim of maximizing the current density in the anode. For this we extracted values of the volumetric current density $i_t(\mathbf{x};\xi_n)$ for each of the 128 inputs ξ_n. The values were taken at 250, 000 locations $\mathbf{x}_j$ and vectorised to form outputs $\mathbf{y}_n$ as described in section 2. The objective is:

$$\xi_* = \underset{\xi}{argmax}\frac{1}{V}\int_{\mathbf{x}} i_t(\mathbf{x};\xi)d\mathbf{x}$$

subject to

$$0.2 \le \xi_1 \le 0.9,\quad 0.1\ \text{V} \le \xi_2 \le 0.95\ \text{V}\quad 800\ \text{K} \le \xi_3 \le 1500\ \text{K},$$
$$0.5\ \text{atm} \le \xi_3 \le 3\ \text{atm} \tag{30}$$

where V is the volume of the anode. We note that the lower and upper bounds on the input are deliberately chosen to fall outside the design space $\mathcal{D}$ used to train the model, so that problem (30) forms a good test for the reliability of the model. We use our multi-fidelity surrogate model for $i_t(\mathbf{x};\xi_n)$, namely, for $\mathbf{y}(\xi)$, with a trapezoidal rule to estimate the integral in (30), based on the components of $\mathbf{y}(\xi)$.

The optimization method used is a primal-dual interior point method, i.e., both the primal and the dual problems are

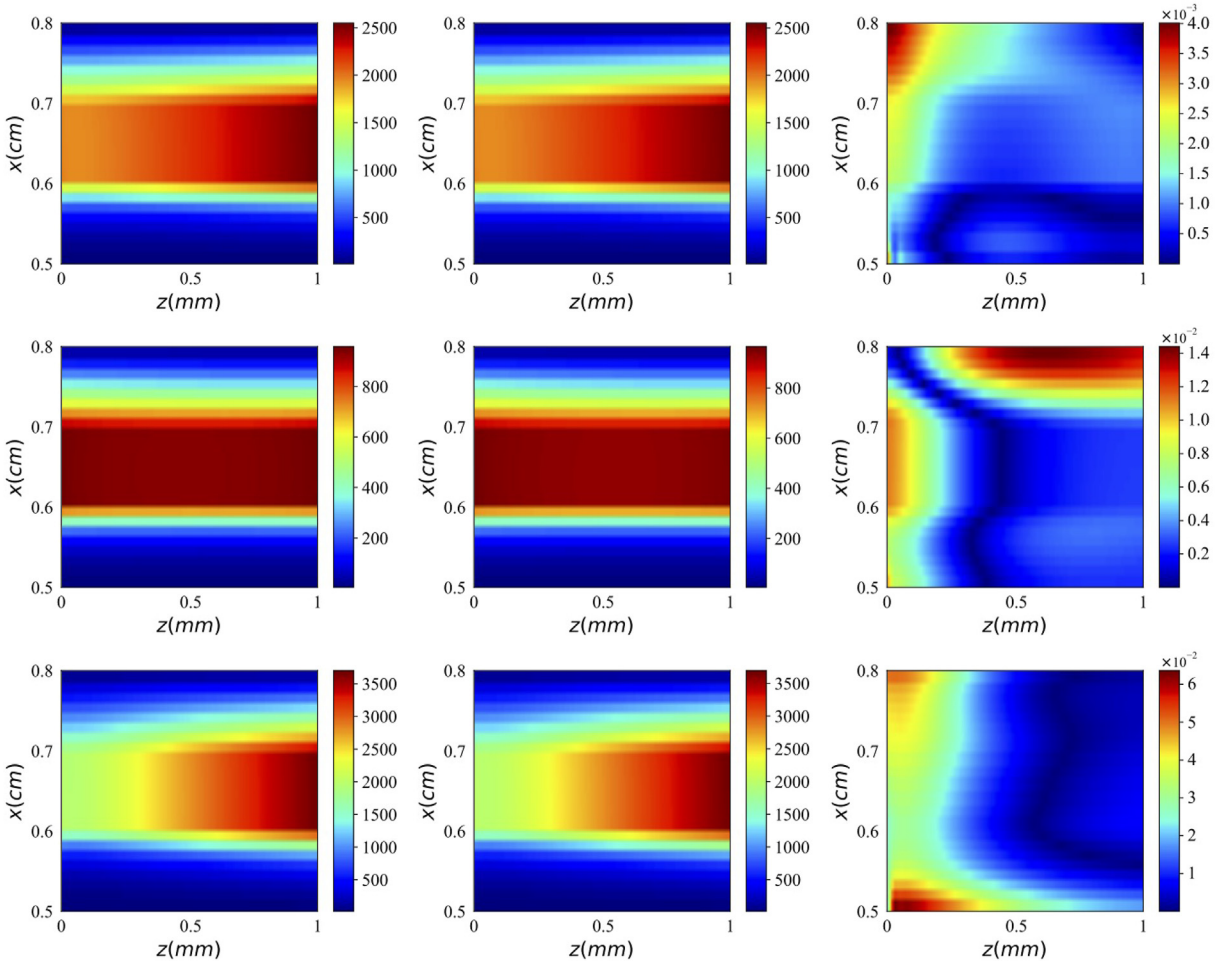

Fig. 7 – Three predictions of the current density $i_e(\mathbf{x};\xi)$ (A m^{-2}) in the $x-z$ plane at the channel centres (Fig. 1) for 80 low-fidelity and 20 high-fidelity training points. These predictions were chosen such that the top row exhibited the lowest square error, the middle row close to the median square error and the bottom row the highest square error. The first column contains the prediction, the second column the tests (ground truths) and the third column the point-wise absolute differences.

simultaneously solved, implemented in Matlab (the 'fmincon' function). This particular interior point method [59] is a variant of predictor-corrector infeasible-interior-point algorithm by Mehrotra [60], which calculates a Newton (prediction) direction, followed by a corrector direction, in which a barrier is introduced.

The work flow is as follows:

1. Load the data set (inputs and outputs for low and high fidelity)
2. Use Eqs. 20 and 21 to obtain the low-fidelity posterior prediction for $\mathbf{y}^L(\xi)$ for a given number of low fidelity training points
3. The high fidelity predictive posterior mean for $\mathbf{y}^H(\xi)$ is given by Eqs. (26) and (27), using Eqs. (11), (21) and (22), for a given number of low and high fidelity training points
4. Perform the optimization using the predictive posterior mean for $\mathbf{y}^H(\xi)$ to estimate the objective function using a trapezoidal rule

Using 80 low-fidelity and 40 high-fidelity training points, the results are summarised in Table 2, averaged over 5 experiments. The average time taken to estimate ξ^* (the entire procedure described above) was 22.95 s, and yielded $\xi^*=$ (0.9,0.1 V, 800 K,3 atm)$^T - \boldsymbol{\epsilon}$, in which each component of $\boldsymbol{\epsilon}$ is $\mathcal{O}(10^{-7})$. The function evaluation count ranged from 87 to 153, underlining the benefit of replacing the high-fidelity solver with the surrogate. The RMSE for the high-fidelity prediction against 88 high-fidelity test points was 1.334×10^{-4} A cm^{-3} per component of $\mathbf{y}(d = 250, 000)$, while the mean value of $i_t(\mathbf{x};\xi_n)$ in the data set was 19.754 A cm^{-3}.

The result in Table 2 is of course correct, since for the highest current density we need to use the highest porosity and channel inlet pressures, together with the lowest cell voltage and temperature. Note that in the COMSOL model, the inlet mole fractions of species are fixed and the channel inlet concentrations C follow the ideal gas law $C = P/(RT)$. Thus, higher temperatures lead to lower concentrations for a fixed channel pressure, which overwhelms any other effects

Table 2 – Solutions to the optimization problems (30)–(33), together with the function evaluation counts and the times taken.

Objective	ξ_1	ξ_2/V	ξ_3/K	ξ_4/atm	Time/s	Function evals
(30)	0.9	0.1	800	3	22.95	121
(31)	0.9	0.1	800	3	12.11	67
(32)	0.2	0.1	800	3	17.27	316
(33)	0.2009	0.6957	1100.4	2.9992	17.27	387

associated with higher temperatures, such as faster kinetics. The highest channel concentrations are attained when $P = 3$ atm and $T = 800$K. Fixing the channel inlet pressures and varying the concentrations independently would lead to the opposite result, namely that higher temperatures would lead to higher current densities.

Applying the same procedure to:

$$\xi_* = \underset{\xi}{argmax} \frac{1}{A} \int_{\mathbf{x}} i_e(\mathbf{x};\xi) d\mathbf{x}$$

subject to

$$0.2 \le \xi_1 \le 0.9, \quad 0.1 \text{ V} \le \xi_2 \le 0.95 \text{ V}, \quad 800 \text{ K} \le \xi_3 \le 1500 \text{ K}, \quad 0.5 \text{ atm} \le \xi_3 \le 3 \text{ atm} \tag{31}$$

in which we maximize the average of the electrolyte current density in the $x - z$ plane (with area A) at the channel centres, led to the same result (to within 10^{-6} for all components of ξ) for the same reasons, as summarised in Table 2.

We now turn our attention to maximizing the magnitude of the average electrolyte potential in the $x - z$ plane at the channel centres:

$$\xi_* = \underset{\xi}{argmax} \left| \frac{1}{A} \int_{\mathbf{x}} \phi_e(\mathbf{x};\xi) d\mathbf{x} \right|$$

subject to

$$0.2 \le \xi_1 \le 0.9, \quad 0.1 \text{ V} \le \xi_2 \le 0.95 \text{ V}, \quad 800 \text{ K} \le \xi_3 \le 1500 \text{ K}, \quad 0.5 \text{ atm} \le \xi_3 \le 3 \text{ atm} \tag{32}$$

The result (to within 10^{-4} for all components of ξ) is shown in Table 2. Here we see that the maximum occurs for the lowest temperature and highest pressure (i.e., the highest channel concentrations), the lowest cell voltage and the lowest porosity. This is as expected, since the lowest electrode porosity leads to the highest volume fraction of electrolyte. In this case the average number of function evaluations was 316, which would take around 35 h to complete with the original high-fidelity COMSOL model. We now add a nonlinear constraint to (32), namely that the minimum value of $\phi_e(\mathbf{x};\xi)$ should not be less than -0.2V:

$$\xi_* = \underset{\xi}{argmax} \left| \frac{1}{A} \int_{\mathbf{x}} \phi_e(\mathbf{x};\xi) d\mathbf{x} \right|$$

subject to

$$-0.2 \text{ V} \le \min_{\xi} \phi_e(\mathbf{x};\xi)$$

$$0.2 \le \xi_1 \le 0.9, \quad 0.1 \text{ V} \le \xi_2 \le 0.95 \text{ V}, \quad 800 \text{ K} \le \xi_3 \le 1500 \text{ K}, \quad 0.5 \text{ atm} \le \xi_3 \le 3 \text{ atm} \tag{33}$$

The result is again shown in Table 2. The values were averaged over 5 runs, with differences between the values of each coordinate of ξ within 0.5%. Here, we find that to maximize the magnitude of the average electrolyte potential, the added constraint raises the optimal temperature to ca. 1100 K and raises the cell voltage to ca. 0.7 V, while the channel pressure and electrode porosity remain unchanged. Both a higher cell voltage and a higher temperature (which lowers the channel concentrations) will lower the rate of charge transfer, and therefore lower the magnitude of the average electrolyte potential.

The final experiment relates to multiple objectives, namely:

$$\xi_* = \underset{\xi}{argmax}\, \mathbf{f}(\xi) = \left(\frac{1}{V} \int_{\mathbf{x}} i_t(\mathbf{x};\xi) d\mathbf{x}, \quad -\frac{1}{V} \int_{\mathbf{x}} \eta_c(\mathbf{x};\xi) d\mathbf{x} \right)^T$$

subject to

$$0.4 \le \xi_1 \le 0.8, \quad 0.2 \text{ V} \le \xi_2 \le 0.85 \text{ V}, \quad 973 \text{ K} \le \xi_3 \le 1273 \text{ K}, \quad 0.5 \text{ atm} \le \xi_3 \le 2.5 \text{ atm} \tag{34}$$

in which we simultaneously wish to maximize the average volumetric anode current density and minimize the magnitude of the average cathode overpotential $\eta_c(\mathbf{x};\xi)$. As with the volumetric anode current density, the cathode overpotential was recorded at 250, 000 spatial locations and vectorised to form outputs $\mathbf{y}_n$. The objectives in this case are clearly competing. To solve this problem we approximate the Pareto front. This is the first stage of a so-called Generate-First Choose-Later approach, in which a set of Pareto optimal solutions is first generated as a candidate set, and from which the design choice can subsequently be made according to other considerations, experience and preferences. A solution is called Pareto optimal or non-dominated if none of the objectives can be improved upon with degrading one or more of the other objectives. The Pareto front is the set of all such solutions, and it was approximated using the genetic algorithm implemented in the 'gamultiobj' function in Matlab.

The average time taken to build the surrogate model and conduct the optimization (over 5 experiments) was 242.43 s, and on average 7350 function evaluations were required. The results are shown in Fig. 8, in which the objective function values corresponding to the Pareto optimal solutions are plotted against each other. In total, a set of 20 Pareto optimal solutions were generated, all of which could be considered as candidates for achieving the objective.

Although, as stated, the objectives chosen are for illustration rather than for any particular physical significance, most of the results above are readily explainable, which facilitates the validation of the approach. Other objectives (single or multiple) would not require a qualitatively different approach. The method is flexible and general enough to incorporate any

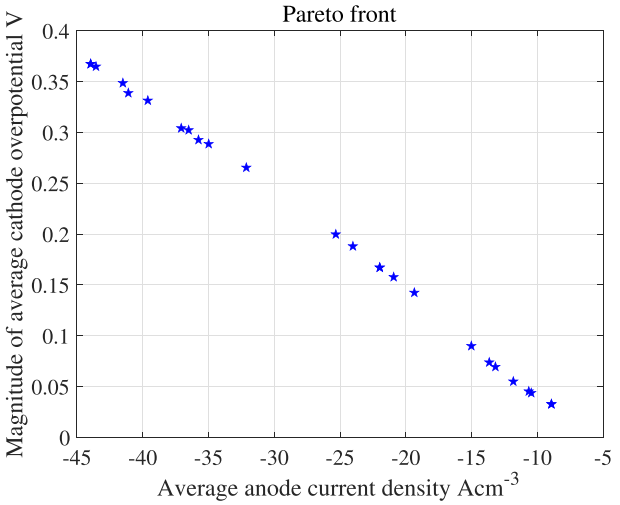

Fig. 8 – A plot of the objective function values corresponding to the Pareto optimal solutions to problem (34).

summary objective or objectives, as long as the corresponding data is available from the multi-fidelity model.

Summary and conclusions

Modelling and simulation play a major role in the development of fuel cell technologies. The complexity of computer models and the attendant times associated with their execution has led to a growth in surrogate modelling approaches for applications such as optimization. Very few such approaches are based on multi-fidelity methods, despite their advantages over pure machine learning models. In addition, few attempts have been made in the fuel cell literature to capture spatial variations in key quantities with surrogate models. In this paper we extended the multi-fidelity stochastic collocation framework for spatially distributed outputs using a feature enhancement, and removed the requirement for exact low-fidelity outputs during inference by using another surrogate model. Our approach is Bayesian, so that estimates of uncertainty are possible.

We showed that the addition of features into the original stochastic colocation formulation can improve its accuracy. Experiments on a detailed SOFC model at two fidelities showed that our approach leads to a highly accurate surrogate model, outperforming several state-of-the-art multi-fidelity methods. It was used in several SOFC optimization problems, for a range of different objectives, for objectives with nonlinear constraints and for multiple objectives. The level of detail furnished by our approach allows for a great deal of flexibility in deciding the objectives a-posteriori, in contrast to focusing on specific scalar targets. It is also worth noting that using the original high-fidelity model would not be feasible for many problems of this nature, such as the multi-objective example we consider, in which over 7000 function evaluations were required.

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

This work was supported by the Creative Research Groups of the National Natural Science Foundation of China (No. 52021004).

REFERENCES

[1] Singh Rathore Shambhu, Biswas Saheli, Fini Daniel, Kulkarni Aniruddha P, Giddey Sarbjit. Direct ammonia solid-oxide fuel cells: a review of progress and prospects. Int J Hydrogen Energy 2021;46(71):35365–84.
[2] Abe JO, Popoola API, Ajenifuja E, Popoola OM. Hydrogen energy, economy and storage: review and recommendation. Int J Hydrogen Energy 2019;44(29):15072–86.
[3] Ghorbani Babak, Krishna Vijayaraghavan. A review study on software-based modeling of hydrogen-fueled solid oxide fuel cells. Int J Hydrogen Energy 2019;44(26):13700–27.
[4] Raj Kamal Abdul Rasheed, Liao Quan, Zhang Caizhi, Chan Siew Hwa. A review on modelling of high temperature proton exchange membrane fuel cells (ht-pemfcs). Int J Hydrogen Energy 2017;42(5):3142–65.
[5] Wang Kun, Hissel Daniel, Péra Marie-Cecile, Steiner Nadia, Marra Dario, Sorrentino Marco, Pianese Cesare, Monteverde Michela, Cardone Pietro, Saarinen Jaakko. A review on solid oxide fuel cell models. Int J Hydrogen Energy 2011;36(12):7212–28.
[6] Shah AA, Luo KH, Ralph TR, Walsh FC. Recent trends and developments in polymer electrolyte membrane fuel cell modelling. Electrochim Acta 2011;56(11):3731–57.
[7] Tzelepis Stefanos, Kavadias Kosmas A, Marnellos George E, George Xydis. A review study on proton exchange membrane fuel cell electrochemical performance focusing on anode and cathode catalyst layer modelling at macroscopic level. Renew Sustain Energy Rev 2021;151:111543.
[8] Li Wei-Zhuo, Yang Wei-Wei, Wang Ning, Jiao Yu-Hang, Yang Yu, Qu Zhi-Guo. Optimization of blocked channel design for a proton exchange membrane fuel cell by coupled genetic algorithm and three-dimensional cfd modeling. Int J Hydrogen Energy 2020;45(35):17759–70.
[9] Antonio Salva J, Iranzo Alfredo, Rosa Felipe, Tapia Elvira, Lopez Eduardo, Isorna Fernando. Optimization of a pem fuel cell operating conditions: obtaining the maximum performance polarization curve. Int J Hydrogen Energy 2016;41(43):19713–23.
[10] Hasanien Hany M, Mohamed A, Shaheen M, Turky Rania A, Qais Mohammed H, Alghuwainem Saad, Kamel Salah, Tostado-Véliz Marcos, Jurado Francisco. Precise modeling of pem fuel cell using a novel enhanced transient search optimization algorithm. Energy 2022;247:123530.
[11] Pourkiaei Seyed Mohsen, Pourfayaz Fathollah, Mehrpooya Mehdi, Ahmadi Mohammad H. Multi-objective optimization of tubular solid oxide fuel cells fed by natural gas: an energetic and exergetic simultaneous optimization. J Therm Anal Calorimetry 2021;145(3):1575–83.

[12] Mojaver Parisa, Khalilarya Shahram, Chitsaz Ata, Assadi Mohsen. Multi-objective optimization of a power generation system based sofc using taguchi/ahp/topsis triple method. Sustain Energy Technol Assessments 2020;38:100674.
[13] Yang Bo, Wang Jingbo, Zhang Mengting, Shu Hongchun, Tao Yu, Zhang Xiaoshun, Yao Wei, Sun Liming. A state-of-the-art survey of solid oxide fuel cell parameter identification: modelling, methodology, and perspectives. Energy Convers Manag 2020;213:112856.
[14] Guo Meiting, Zhao Dongqi, Xu Qidong, Zheng Li, Xu Haoran, Ni Meng. New interconnector designs for electrical performance enhancement of solid oxide fuel cells: a 3d modelling study. J Power Sources 2022;533:231373.
[15] Guo Meiting, Zhao Dongqi, Xu Qidong, Zheng Li, Xu Haoran, Ni Meng. New interconnector design optimization to balance electrical and mechanical performance of solid oxide fuel cell stack. Int J Hydrogen Energy 2023;48(8):3107–21.
[16] Abdullah Taufiq, Liu Lin. Simulation-based microstructural optimization of solid oxide fuel cell for low temperature operation. Int J Hydrogen Energy 2016;41(31):13632–43.
[17] Kriston Akos, Podias Andreas, Adanouj Ibtissam, Andreas Pfrang. Analysis of the effect of thermal runaway initiation conditions on the severity of thermal runaway—numerical simulation and machine learning study. J Electrochem Soc 2020;167(9):090555.
[18] Radaideh Majdi I, Radaideh Mohammed I, Kozlowski Tomasz. Design optimization under uncertainty of hybrid fuel cell energy systems for power generation and cooling purposes. Int J Hydrogen Energy 2020;45(3):2224–43.
[19] Pourrahmani Hossein, Siavashi Majid, Moghimi Mahdi. Design optimization and thermal management of the pemfc using artificial neural networks. Energy 2019;182:443–59.
[20] Vichard L, Harel F, Ravey A, Venet P, Hissel D. Degradation prediction of pem fuel cell based on artificial intelligence. Int J Hydrogen Energy 2020;45(29):14953–63.
[21] Sun Ke, Esnaola Iñaki, Okorie Okechukwu, Charnley Fiona, Moreno Mariale, Tiwari Ashutosh. Data-driven modeling and monitoring of fuel cell performance. Int J Hydrogen Energy 2021;46(66):33206–17.
[22] Zhu Jiangong, Wang Yixiu, Huang Yuan, Bhushan Gopaluni R, Cao Yankai, Heere Michael, Mühlbauer Martin J, Mereacre Liuda, Dai Haifeng, Liu Xinhua, et al. Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation. Nat Commun 2022;13(1):1–10.
[23] Li Wenhua, Jiao Zhipeng, Le Du, Fan Wenyi, Zhu Yazun. An indirect rul prognosis for lithium-ion battery under vibration stress using elman neural network. Int J Hydrogen Energy 2019;44(23):12270–6.
[24] Guarino Antonio, Trinchero Riccardo, Canavero Flavio, Spagnuolo Giovanni. A fast fuel cell parametric identification approach based on machine learning inverse models. Energy 2022;239:122140.
[25] Ritzberger Daniel, Höflinger Johannes, Du Zhang Peng, Hametner Christoph, Jakubek Stefan. Data-driven parameterization of polymer electrolyte membrane fuel cell models via simultaneous local linear structured state space identification. Int J Hydrogen Energy 2021;46(21):11878–93.
[26] Peksen M, Blum L, Stolten D. Optimisation of a solid oxide fuel cell reformer using surrogate modelling, design of experiments and computational fluid dynamics. Int J Hydrogen Energy 2012;37(17):12540–7.
[27] Wilberforce Tabbi, Olabi AG. Proton exchange membrane fuel cell performance prediction using artificial neural network. Int J Hydrogen Energy 2021;46(8):6037–50.
[28] Milewski Jarosław, Świrski Konrad. Modelling the sofc behaviours by artificial neural network. Int J Hydrogen Energy 2009;34(13):5546–53.
[29] Subotić Vanja, Eibl Michael, Hochenauer Christoph. Artificial intelligence for time-efficient prediction and optimization of solid oxide fuel cell performances. Energy Convers Manag 2021;230:113764.
[30] Zhang Xuexia, Zhou Jingzhe, Chen Weirong. Data-driven fault diagnosis for pemfc systems of hybrid tram based on deep learning. Int J Hydrogen Energy 2020;45(24):13483–95.
[31] Lu Jiahuan, Xiong Rui, Tian Jinpeng, Wang Chenxu, Hsu Chia-Wei, Tsou Nien-Ti, Sun Fengchun, Ju Li. Battery degradation prediction against uncertain future conditions with recurrent neural network enabled deep learning. Energy Storage Mater 2022;50:139–51.
[32] Choi Jaeyoo, Cha Yohan, Kong Jihoon, Vaz Neil, Lee Jaeseung, Ma Sang-Bum, Kim Jin-Hyuk, Lee Seung Woo, Jang Seung Soon, Ju Hyunchul. Multi-variate optimization of polymer electrolyte membrane fuel cells in consideration of effects of gdl compression and intrusion. J Electrochem Soc 2022;169(1):014511.
[33] Xu Haoran, Ma Jingbo, Tan Peng, Chen Bin, Wu Zhen, Zhang Yanxiang, Wang Huizhi, Jin Xuan, Ni Meng. Towards online optimisation of solid oxide fuel cell performance: combining deep learning with multi-physics simulation. Energy and AI 2020;1:100003.
[34] Xu Guoping, Yu Zeting, Xia Lei, Wang Changjiang, Ji Shaobo. Performance improvement of solid oxide fuel cells by combining three-dimensional cfd modeling, artificial neural network and genetic algorithm. Energy Convers Manag 2022;268:116026.
[35] Wang Yu, Jiang Wenchun, Luo Yun, Song Ming, Tu Shan-Tung. High temperature creep strength design and optimization of solid oxide fuel cell. Int J Hydrogen Energy 2022;47(50):21450–61.
[36] Shah AA. Surrogate modeling for spatially distributed fuel cell models with applications to uncertainty quantification. Journal of Electrochemical Energy Conversion and Storage 2017;14(1).
[37] Xing WW, Triantafyllidis V, Shah AA, Nair PB, Zabaras Nicholas. Manifold learning for the emulation of spatial fields from computational models. J Comput Phys 2016;326:666–90.
[38] Xing WW, Yu F, Leung PK, Li X, Wang P, Shah AA. A new multi-task learning framework for fuel cell model outputs in high-dimensional spaces. J Power Sources 2021;482:228930.
[39] Giselle Fernandez-Godino M, Park Chanyoung, Kim Nam-Ho, Haftka Raphael T. Review of multi-fidelity models. 2016.
[40] Liu Yingxu, Dirkes Steffen, Kohrn Markus, Wick Maximilian, Pischinger Stefan. A high-fidelity real-time capable dynamic discretized model of proton exchange membrane fuel cells for the development of control strategies. J Power Sources 2022;537:231394.
[41] Voskuilen Tyler G, Moffat Harry K, Schroeder Benjamin B, Roberts Scott A. Multi-fidelity electrochemical modeling of thermally activated battery cells. J Power Sources 2021;488:229469.
[42] Xing WW, Shah AA, Wang P, Zhe S, Fu Q, Kirby RM. Residual Gaussian process: a tractable nonparametric bayesian emulator for multi-fidelity simulations. Appl Math Model 2021;97:36–56.
[43] Kennedy Marc C, Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. Biometrika 2000;87(1):1–13.
[44] Le Gratiet Loic. *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis. Université Paris-Diderot-Paris VII; 2013.
[45] Paris Perdikaris, Raissi Maziar, Damianou Andreas, Lawrence ND, Karniadakis George Em. Nonlinear

information fusion algorithms for data-efficient multi-fidelity modelling. Proc R Soc A 2017;473(2198):20160751.
[46] Parussini L, Venturi D, Perdikaris P, Karniadakis GE. Multi-fidelity Gaussian process regression for prediction of random fields. J Comput Phys May 2017;336(C):36–50.
[47] Narayan Akil, Gittelson Claude, Xiu Dongbin. A stochastic collocation algorithm with multifidelity models. SIAM J Sci Comput 2014;36(2):A495–521.
[48] Xing W, Razi M, Kirby RM, Sun K, Shah AA. Greedy nonlinear autoregression for multifidelity computer models at different scales. Energy and AI 2020;1:100012.
[49] Hampton Jerrad, Fairbanks Hillary, Narayan Akil, Doostan Alireza. Parametric/stochastic model reduction: low-rank representation, non-intrusive bi-fidelity approximation, and convergence analysis. arXiv 2017. preprint arXiv:1709.03661.
[50] Zhu Xueyu, Narayan Akil, Xiu Dongbin. Computational aspects of stochastic collocation with multifidelity models. SIAM/ASA J Uncertain Quantification 2014;2(1):444–63.
[51] 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 2018;152(C):125–33.
[52] Skinner Ryan, Doostan Alireza, Peters Eric, Evans John, Jansen Kenneth E. An evaluation of bi-fidelity modeling efficiency on a general family of NACA airfoils. In: 35th AIAA applied aerodynamics conference; 2017. p. 3260.
[53] Razi Mani, Kirby Robert M, Narayan Akil. Fast predictive multi-fidelity prediction with models of quantized fidelity levels. J Comput Phys 2019;376:992–1008.
[54] Jofre Lluis, Geraci Gianluca, Fairbanks Hillary, Doostan Alireza, Iaccarino Gianluca. Multi-fidelity uncertainty quantification of irradiated particle-laden turbulence. arXiv 2018. preprint arXiv:1801.06062.
[55] Hampton Jerrad, Fairbanks Hillary R, Narayan Akil, Doostan Alireza. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. J Comput Phys 2018;368:315–32.
[56] Lunardi Alessandra. Interpolation theory, ume 9. Springer; 2009.
[57] Conti S, O'Hagan A. Bayesian emulation of complex multi-output and dynamic computer models. Journal of Statististical Planning and Inference 2010;140:640–51.
[58] Rasmussen Carl Edward, Williams Christopher KI. Gaussian processes for machine learning. MIT Press; 2006.
[59] Zhang Yin. Solving large-scale linear programs by interior-point methods under the matlab environment. Optim Methods Software 1998;10(1):1–31.
[60] Mehrotra Sanjay. On the implementation of a primal-dual interior point method. SIAM J Optim 1992;2(4):575–601.