Data Fusion With Latent Map Gaussian Processes

Jonathan Tammer Eweis-Labolle
Department of Mechanical and Aerospace Engineering, University of California, Irvine, Irvine, CA 92697
e-mail: jeweisla@uci.edu

Nicholas Oune
Department of Mechanical and Aerospace Engineering, University of California, Irvine, Irvine, CA 92697
e-mail: ounen@uci.edu

Ramin Bostanabad[1]
Department of Mechanical and Aerospace Engineering, University of California, Irvine, Irvine, CA 92697
e-mail: raminb@uci.edu

Multi-fidelity modeling and calibration are data fusion tasks that ubiquitously arise in engineering design. However, there is currently a lack of general techniques that can jointly fuse multiple data sets with varying fidelity levels while also estimating calibration parameters. To address this gap, we introduce a novel approach that, using latent-map Gaussian processes (LMGPs), converts data fusion into a latent space learning problem where the relations among different data sources are automatically learned. This conversion endows our approach with some attractive advantages such as increased accuracy and reduced overall costs compared to existing techniques that need to take a combinatorial approach to fuse multiple datasets. Additionally, we have the flexibility to jointly fuse any number of data sources and the ability to visualize correlations between data sources. This visualization allows an analyst to detect model form errors or determine the optimum strategy for high-fidelity emulation by fitting LMGP only to the sufficiently correlated data sources. We also develop a new kernel that enables LMGPs to not only build a probabilistic multi-fidelity surrogate but also estimate calibration parameters with quite a high accuracy and consistency. The implementation and use of our approach are considerably simpler and less prone to numerical issues compared to alternate methods. Through analytical examples, we demonstrate the benefits of learning an interpretable latent space and fusing multiple (in particular more than two) sources of data. [DOI: 10.1115/1.4054520]

Keywords: Gaussian processes, data fusion, calibration, emulation, manifold learning, machine learning, metamodeling, uncertainty modeling

1 Introduction

Computer models are increasingly employed in the analysis and design of complex systems. For a particular system, there are typically various models available whose fidelity is generally related to their costs; i.e., accurate models are generally more expensive. In such a scenario, *multi-fidelity modeling* techniques are adopted to balance costs and accuracy when using all these models in the analyses [1,2]. Additionally, computer models typically have some *calibration* parameters which are estimated by systematically comparing their predictions to experiments/observations [3]. These parameters either correspond to some properties of the underlying system being modeled or act as tuning knobs that compensate for the model deficiencies. In this paper, we introduce a versatile, efficient, and unified approach for emulation-based multi-fidelity modeling and calibration (henceforth, we use the term data fusion to refer to both multi-fidelity modeling and calibration because they all involve fusing or assimilating multiple sources of data). Our approach is based on latent-map Gaussian processes and its core idea is to convert data fusion into a learning process where different data sources are related in a nonlinearly learned manifold.

Over the past few decades, many data fusion techniques have been developed for outer-loop applications such as design optimization, sequential sampling, or inverse parameter estimation. For example, multi-fidelity modeling can be achieved via space mapping [4–6] or multi-level [7–9] techniques where the inputs of the low-fidelity data are mapped via $\boldsymbol{x}_l = \boldsymbol{F}(\boldsymbol{x}_h)$. In this equation, $\boldsymbol{x}_l$ and $\boldsymbol{x}_h$ are the inputs of low- and high-fidelity data sources, respectively, and $\boldsymbol{F}(\cdot)$ is the transformation function whose *predefined* functional form is calibrated such that $y_l(\boldsymbol{F}(\boldsymbol{x}_h))$ approximates $y_h(\boldsymbol{x}_h)$ as closely as possible. These techniques are particularly useful in applications where higher fidelity data are obtained by successively refining the discretization of the simulation domain [7,9], e.g., by refining the mesh when simulating the flow over an airfoil. The main disadvantage of space mapping techniques is that choosing a near-optimal functional form for $\boldsymbol{F}(\cdot)$ is iterative and very cumbersome.

Two of the most important aspects of multi-fidelity modeling are choosing the emulators that surrogate the data sources and formulating the relation between these emulators. Correspondingly, several methods have been developed based on Gaussian processes (GPs) [3], Co-Kriging [10], polynomial chaos expansions [11,12], and moving least squares [13]. The interested reader is referred to Refs. [2,14] for more comprehensive reviews on multi-fidelity modeling and how they benefit outer-loop applications.

Multi-fidelity modeling is closely related to the calibration of computer models since the latter also involves working with at least two data sources where typically the low-fidelity one possesses the calibration parameters. Besides the traditional ways of estimation that are ad hoc and involve trial and error, there are more systematic methods that are based on generalized likelihood [15] or Bayesian principles [16].

Among existing methods for multi-fidelity modeling and calibration, the most popular emulator-based method in engineering design is that of Kennedy and O'Hagan (KOH) [3] which assimilates and emulates two data sources while estimating calibration parameters of the low-fidelity source (if there are any such parameters). KOH's approach is one of the first attempts that considers a broad range of uncertainty sources arising during the calibration and subsequent uses of the emulator. This approach has been used in many applications including climate simulations [17], materials modeling [18], and modeling shock hydrodynamics [19].

KOH's approach assumes that the discrepancies between the two data sources are additive[2] and that both data sources and the discrepancy between them can be modeled via GPs. The approach then uses (fully [20,21] or modular [18,22–25]) Bayesian inference to find the posterior estimates of the GPs as well as the calibration

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received December 1, 2021; final manuscript received April 28, 2022; published online June 13, 2022. Assoc. Editor: Sayan Ghosh.

[2]Multiplicative terms have also been introduced to KOH's approach but are seldom adopted as they increase the identifiability issues and computational costs while negligibly improving the mean prediction accuracy.

parameters. The fully Bayesian version of KOH's method offers advantages such as low computational costs for small data sets or quantifying various uncertainty sources (e.g., lack of data, noise, model form error, and unknown simulation parameters). However, obtaining the joint posteriors via Markov chain Monte Carlo (MCMC) is quite effortful and expensive, especially in high dimensions or with relatively large datasets. The modular version of KOH's approach addresses this limitation by typically using point estimates for the GP hyperparameters of the low-fidelity data [3,23]. These estimates are obtained via maximum likelihood estimation (MLE) and, while they result in a small under-estimation of uncertainties with small data, provide accurate mean predictions.

A major limitation of KOH's approach and other reviewed data fusion techniques is that they only accommodate two data sources at a time. That is, the fusion process must be repeated p times if there are p low-fidelity and one high-fidelity data sources. In addition to being tedious and expensive, this repetitive process does not provide a straightforward diagnostic mechanism for comparing the low-fidelity sources to identify, e.g., which one(s) perform similarly or have the smallest model form error.

In this paper, we aim to address the abovementioned limitations of the existing technologies for data fusion. Our primary contributions are threefold and summarized as follows. First, we convert multi-fidelity modeling into a latent space learning problem. This conversion is achieved via latent-map Gaussian processes (LMGPs) and endows our approach with important advantages such as flexibility to jointly fuse any number of data sources and ability to visualize correlations between them. This visualization provides the user with an easy-to-interpret diagnostic measure for identifying the relations between different data sources. We believe the joint fusion (of more than two sources) and the accompanying visualization aids reduce the overall costs of multi-fidelity modeling compared to reviewed methods since they eliminate the iterative process of data source selection and link the fusion results across the iterations (note that our approach is also applicable to problems with two data sources). Second, we develop a new kernel function that enables LMGPs to not only build a probabilistic multi-fidelity surrogate but also estimate calibration parameters with high accuracy and consistency. Third, the implementation of our approach is considerably simpler and less prone to numerical issues compared to the reviewed technologies (especially KOH's approach).

The rest of the paper is organized as follows. In Sec. 2, we briefly review the relevant technical background on GPs and LMGPs (see Sec. 7 for Nomenclature). In Sec. 3, we introduce our approach to multi-fidelity modeling and calibration while demonstrating its performance on four pedagogical examples. In Sec. 4, we validate our approach against GPs and KOH's method on six analytic and engineering examples. We conclude the paper in Sec. 5 by discussing the advantages and limitations of our approach, considerations that should be made in its application, and its application to multi-response problems.

2 Emulation via Latent-Map Gaussian Processes

We review emulation via GPs and a variation of GPs (i.e., LMGP) for data sets that include categorical inputs. Throughout, symbols or numbers enclosed in parentheses encode sample numbers and are used either as subscripts or as superscripts. For example, $\boldsymbol{x}_{(i)}$ or $\boldsymbol{x}^{(i)}$ denote the ith sample in a training data set while x_i indicates the ith component of the vector $\boldsymbol{x} = [x_1,\ x_2,\ \ldots, x_{d_x}]^T$. We use h and l either as superscript or as subscript to denote high- and low-fidelity data sources. For instance, $\boldsymbol{x}_h^{(i)}$ and $y_h^{(i)}$ denote, respectively, the inputs and output of the ith sample in the high-fidelity data set. In cases where there is more than one low-fidelity source, we add a number to the l symbol, e.g., $y_{l_3}(\boldsymbol{x})$ denotes the third low-fidelity source. Lastly, we distinguish between the data source (or the underlying function) and samples by specifying the functional dependence (e.g., $y(\boldsymbol{x})$ is a function while y and $\boldsymbol{y}$ are, respectively, a scalar and a vector of values).

Denote the inputs and outputs of a system by d_x-dimensional vector $\boldsymbol{x} = [x_1,\ x_2,\ \ldots, x_{d_x}]^T$ and the scalar y. Assume the training data come from a realization of a GP defined as $\eta(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})\boldsymbol{\beta} + \xi(\boldsymbol{x})$ where $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_h(\boldsymbol{x})]$ are a set of pre-determined parametric functions and $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_h]^T$ are the unknown coefficients. $\xi(\boldsymbol{x})$ is a zero-mean GP whose parameterized covariance function is

$$cov(\xi(\boldsymbol{x}), \xi(\boldsymbol{x}')) = c(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 r(\boldsymbol{x}, \boldsymbol{x}') \tag{1}$$

where σ^2 is the process variance and $r(\cdot, \cdot)$ is a user-defined parametric correlation function. There are many types of correlation functions [26,27], but the most common one is the Gaussian kernel

$$r(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\sum_{i=1}^{d_x} 10^{\omega_i}(x_i - x_i')^2\right\} = \exp((\boldsymbol{x} - \boldsymbol{x}')^T \Omega_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}')) \tag{2}$$

where $\boldsymbol{\omega} = [\omega_1,\ \ldots,\ \omega_{d_x}]^T$, $-\infty < \omega_i < \infty$ are the roughness or scale parameters (in practice the ranges are limited to $-10 < \omega_i < 6$ ensure numerical stability [26,28]) and $\Omega_{\boldsymbol{x}} = diag(10^{\boldsymbol{\omega}})$.

The correlation function in Eq. (2) depends on the distance between two arbitrary input points $\boldsymbol{x}$ and $\boldsymbol{x}'$. Hence, traditional GPs cannot accommodate categorical inputs (such as gender and zip code) as they do not possess a distance metric. This issue is well established in the literature, and there exist a number of strategies that address it by reformulating the covariance function such that it can handle categorical variables [29–32]. In this paper, we use LMGPs [33] which are recently developed and shown to outperform previous methods.

Let us denote the categorical inputs by $\boldsymbol{t} = [t_1, \ldots,\ t_{d_t}]^T$ where the total number of distinct levels for qualitative variable t_i is m_i. For instance, $t_1 = \{92697, 92093\}$ and $t_2 = \{math, physics, chemistry\}$ are two categorical inputs that encode zip code ($m_1 = 2$ levels) and course subject ($m_2 = 3$ levels), respectively. Inputs for mixed (numerical and categorical) training data are collectively denoted by $\boldsymbol{u} = [\boldsymbol{x}; \boldsymbol{t}]$, which is a column vector of size $(d_x + d_t) \times 1$. To handle mixed inputs, LMGP maps categorical variables to some points in a manifold. This mapping allows using any standard correlation function such as the Gaussian which is reformulated as follows:

$$r(\boldsymbol{u}, \boldsymbol{u}') = \exp\{-\|\boldsymbol{z}(\boldsymbol{t}) - \boldsymbol{z}(\boldsymbol{t}')\|_2^2 - (\boldsymbol{x} - \boldsymbol{x}')^T \Omega_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}')\} \tag{3}$$

where $\|\cdot\|_2$ denotes the Euclidean 2-norm and $\boldsymbol{z}(\boldsymbol{t}) = [z_1(\boldsymbol{t}), \ldots, z_{d_z}(\boldsymbol{t})]_{1 \times d_z}$ is the to-be-learned latent space point corresponding to the particular combination of the categorical variables denoted by $\boldsymbol{t}$. To find these points in the latent space, LMGP first assigns a unique vector (i.e., a prior representation) to each combination of categorical variables. Then, it uses matrix multiplication to map each of these vectors to a point in a manifold of dimensionality d_z

$$\boldsymbol{z}(\boldsymbol{t}) = \boldsymbol{\zeta}(\boldsymbol{t})\boldsymbol{A} \tag{4}$$

where $\boldsymbol{\zeta}(\boldsymbol{t})$ is the $1 \times \sum_{i=1}^{d_t} m_i$ unique prior vector representation of $\boldsymbol{t}$ and $\boldsymbol{A}$ is a $\sum_{i=1}^{d_t} m_i \times d_z$ matrix that maps $\boldsymbol{\zeta}(\boldsymbol{t})$ to $\boldsymbol{z}(\boldsymbol{t})$. In this paper, we use $d_z = 2$ since it simplifies visualization and has also been shown to provide sufficient flexibility for learning the latent relations [33].

We can construct $\boldsymbol{\zeta}$ in a number of ways, see Ref. [33] for more information on selecting the priors. In this paper, we use a form of one-hot-encoding. Specifically, we first construct the $1 \times m_i$ vector $\boldsymbol{\nu}^i = [\nu_1^i,\ \nu_2^i,\ \ldots,\ \nu_{m_i}^i]$ for the categorical variable t_i such that $\nu_j^i = 1$ when t_i is at level j and $\nu_j^i = 0$ when t_i is at level $k \neq j$ for, $k \in 1, 2, \cdots, m_i$. Then, we set $\boldsymbol{\zeta}(\boldsymbol{t}) = [\boldsymbol{\nu}^1,\ \boldsymbol{\nu}^2,\ \cdots,\ \boldsymbol{\nu}^{d_t}]$. For instance, in the above example with two categorical variables, $t_1 = \{92697, 92093\}$ and $t_2 = \{math, physics, chemistry\}$, we encode the

combination $\boldsymbol{t} = [92093, physics]^T$ by $\boldsymbol{\zeta}(\boldsymbol{t}) = [0, 1, 0, 1, 0]$ where the first two elements encode zip code while the rest encode the subject.

To emulate via LMGP, point estimates of $\boldsymbol{A}, \boldsymbol{\beta}, \boldsymbol{\omega}$, and σ^2 must be determined based on the data. These estimates can be found via either cross-validation (CV) or MLE. Alternatively, Bayes' rule can be applied to find posterior distributions of the hyperparameters if prior knowledge is available. In this paper, MLE is employed because it provides a high generalization power while minimizing the computational costs [27,34]. MLE works by estimating $\boldsymbol{A}$, $\boldsymbol{\beta}$, $\boldsymbol{\omega}$, and σ^2 such that they maximize the likelihood of n training data being generated by $\eta(\boldsymbol{x})$. This optimization can be equivalently expressed as

$$[\hat{\boldsymbol{\beta}},\ \hat{\sigma}^2,\ \hat{\boldsymbol{\omega}}, \hat{\boldsymbol{A}}] = \underset{\boldsymbol{\beta},\, \sigma^2,\, \boldsymbol{\omega},\, \boldsymbol{A}}{\operatorname{argmin}} \ \frac{n}{2}\log(\sigma^2) + \frac{1}{2}\log(|\boldsymbol{R}|) + \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{5}$$

where $\boldsymbol{R}$ and $\hat{\sigma}^2$ are now functions of both $\boldsymbol{\omega}$ and $\boldsymbol{A}$, $\log(\cdot)$ is the natural logarithm, $|\cdot|$ denotes the determinant operator, $\boldsymbol{y} = [y_{(1)}, \ldots, y_{(n)}]^T$ is the $n \times 1$ vector of outputs in the training data, $\boldsymbol{R}$ is the $n \times n$ correlation matrix with the (i, j)th element $R_{ij} = r(\boldsymbol{x}_{(i)}, \boldsymbol{x}_{(j)})$ for $i, j = 1, \ldots, n$, and $\boldsymbol{F}$ is the $n \times h$ matrix with the (k, l)th element $F_{kl} = f_l(\boldsymbol{x}_{(k)})$ for $k = 1, \ldots, n$ and $l = 1, \ldots, h$. By setting the partial derivatives with respect to $\boldsymbol{\beta}$ and σ^2 to zero, their estimates can be solved in terms of $\boldsymbol{\omega}$ and $\boldsymbol{A}$ as follows:

$$\hat{\boldsymbol{\beta}} = [\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F}]^{-1}[\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{y}] \tag{6}$$

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}) \tag{7}$$

Plugging these estimates into Eq. (5) and removing the constants yield:

$$[\hat{\boldsymbol{\omega}},\ \hat{\boldsymbol{A}}] = \underset{\boldsymbol{\omega},\, \boldsymbol{A}}{\operatorname{argmin}} \ n\log(\hat{\sigma}^2) + \log(|\boldsymbol{R}|) = \underset{\boldsymbol{\omega},\, \boldsymbol{A}}{\operatorname{argmin}}\, L \tag{8}$$

By minimizing L one can solve for $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{\omega}}$ subsequently obtain $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ using Eqs. (6) and (7). While many heuristic global optimization methods exist such as genetic algorithms [35] and particle swarm optimization [36], gradient-based optimization techniques based on, e.g., the L-BFGS algorithm [37], are generally preferred due to their ease of implementation and superior computational efficiency [26,38]. With gradient-based approaches, it is essential to start the optimization via numerous initial guesses to improve the chances of achieving global optimality [33,38].

After obtaining the hyperparameters via MLE, the response at any $\boldsymbol{x}^*$ is estimated via $\mathbb{E}[y^*] = \boldsymbol{f}(\boldsymbol{x}^*)\hat{\boldsymbol{\beta}} + \boldsymbol{g}^T(\boldsymbol{x}^*)\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})$ where $\mathbb{E}$ denotes expectation, $\boldsymbol{f}(\boldsymbol{x}^*) = [f_1(\boldsymbol{x}^*), \ \ldots, f_h(\boldsymbol{x}^*)]$, $\boldsymbol{g}(\boldsymbol{x}^*)$ is an $n \times 1$ vector with the ith element $c(\boldsymbol{x}_{(i)}, \boldsymbol{x}^*) = \hat{\sigma}^2 r(\boldsymbol{x}_{(i)}, \ \boldsymbol{x}^*)$, and $\boldsymbol{V}$ is the covariance matrix with the (i, j)th element $\hat{\sigma}^2 r(\boldsymbol{x}_{(i)}, \ \boldsymbol{x}_{(j)})$. Additionally, The posterior covariance between the responses at the two inputs $\boldsymbol{x}^*$ and $\boldsymbol{x}'$ is $cov(y^*, y') = c(\boldsymbol{x}^*, \boldsymbol{x}') - \boldsymbol{g}^T(\boldsymbol{x}^*)\boldsymbol{V}^{-1}\boldsymbol{g}(\boldsymbol{x}') + \boldsymbol{h}(\boldsymbol{x}^*)(\boldsymbol{F}^T\boldsymbol{V}^{-1}\boldsymbol{F})^{-1}\boldsymbol{h}(\boldsymbol{x}')^T$ where $\boldsymbol{h}(\boldsymbol{x}^*) = (\boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{F}^T\boldsymbol{V}^{-1}\boldsymbol{g}(\boldsymbol{x}^*))$.

The above formulations can be easily extended to cases where the data set is noisy. GPs (and hence LMGPs) can address noise and smoothen data by using a nugget or jitter parameter, δ, which is incorporated into the correlation matrix. That is, $\boldsymbol{R}$ becomes $\boldsymbol{R}_\delta = \boldsymbol{R} + \delta\boldsymbol{I}_{n\times n}$ where $\boldsymbol{I}_{n\times n}$ is the identity matrix of size $n \times n$. If the nugget parameter is used, the estimated (stationary) noise variance in the data will be $\delta\hat{\sigma}^2$. The version of LMGP used in this paper finds only one nugget parameter and uses it for all categorical combinations; i.e., we assume that the noise level is the same for each data set. LMGP can be modified in a straightforward manner to have a separate nugget parameter (and hence separate noise estimate) for each categorical combination.

3 Proposed Framework for Data Fusion

In this section, we first explain the core idea and rationale of our approach in Sec. 3.1. Then, we detail how it is used for multi-fidelity modeling and calibration in Secs. 3.2 and 3.3, respectively. In the latter two subsections, we provide pedagogical examples to facilitate the discussions and elaborate on the benefits of the learned latent space in diagnosing the results. The notation introduced in Sec. 2 is also used here (see Sec. 7 for Nomenclature).

3.1 The Rationale Behind Using a Latent Space for Data Fusion. Factors that affect the fidelity of various data sources are either known or not; in either case, they typically cannot be easily used in the fusion process. Consider an engineering application on predicting the fracture toughness of an alloy where an engineer states "model A and model B achieve errors of 7% and 12% when their predictions are tested against experimental data." These inaccuracies and their 5% difference can be due to many underlying factors such as noise in the experiments, missing physics in either of the models (especially model B), uncertain material properties (i.e., calibration parameters) that affect the fracture behavior, or numerical errors associated with the computer models (e.g., coarse discretization). It is very difficult to quantitatively incorporate all these factors into data fusion. Hence, existing fusion methods such as that of the Kennedy and O'Hagan [3] assign *labels* or *qualitative* variables to data (e.g., data from "model A" or data from "experiments") and then develop fusion formulas that break down if the underlying assumptions are incorrect or if there are many information sources.

We argue that data fusion should be based on *learned quantitative* variables instead of assigned qualitative labels to enable instruction-free and versatile fusion. We use LMGPs to learn these quantitative variables (other methods can be used as well) in a latent space that aims to encode the underlying factors which distinguish different data sources. The power of latent spaces in learning hidden factors is perhaps best exemplified in computer vision where deep neural networks encode high-dimensional images to a low-dimensional latent space where a single axis learns *smiling* (Fig. 1(*a*)).

As shown in Fig. 1(*b*), data fusion via LMGP is achieved via the following steps. First, we augment the various datasets with categorical inputs that aim to distinguish the data sources and also add unknown calibration parameters (if applicable). Then, we fit a single LMGP to the combined data set to obtain emulators of the data sources and estimates of the calibration parameters (if applicable). Finally, once the LMGP is trained, we visualize the learned latent space to analyze the relations between the sources. In the following subsections, we provide more details on each of these steps.

Following the above--mentioned description, we summarize our goals in data fusion as building emulators for each data source (especially the high-fidelity one), estimating any unknown calibration parameters, and automatically obtaining the relation between the various data sources. We also note that our approach can simultaneously fuse any number of data sources with any level of fidelity. Without lack of generality, hereafter, we will assign only one source as high fidelity and the rest of the sources are treated as low fidelity. This assignment is adopted to simplify the descriptions and does *not* affect our approach at all since we do not use any knowledge on the fidelity level during fusion (e.g., if there are two experimental and three simulation data sets, we can assign any one of them as high-fidelity and the rest as low-fidelity).

We also assume that the goal of the problem is to emulate the data source with the highest fidelity level, which entails emulation of the true system, and to estimate the best calibration parameters, if applicable. To measure the accuracy of $y_{l_i}(\boldsymbol{x})$ with respect to $y_h(\boldsymbol{x})$, we evaluate relative root-mean-squared error (RRMSE)

$$\text{RRMSE}\ (y_{l_i}(\boldsymbol{x})) \approx \sqrt{\frac{(\boldsymbol{y}_{l_i} - \boldsymbol{y}_h)^T(\boldsymbol{y}_{l_i} - \boldsymbol{y}_h)}{n \times \text{var}(\boldsymbol{y}_h)}} \tag{9}$$

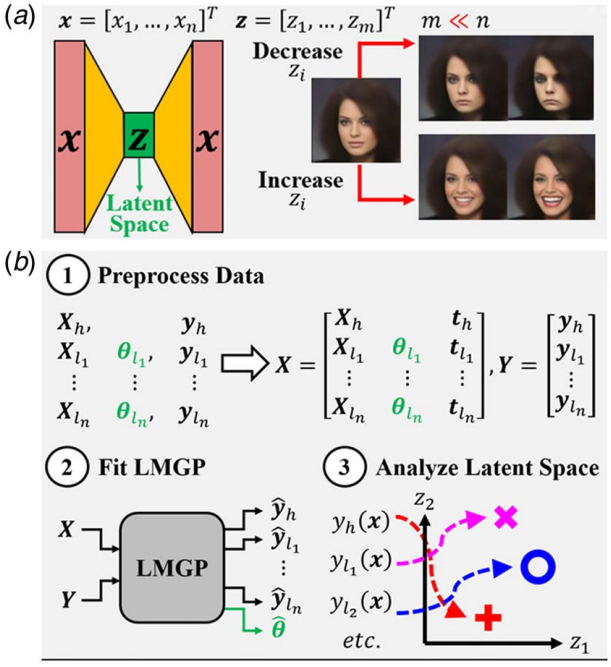

Fig. 1 Data fusion as a latent space learning problem: (*a*) latent representation of facial features: A latent representation enables drastic reduction of dimensionality such that each axis encodes a complex feature. (*b*) Data fusion with LMGP: Calibration inputs and outputs, denoted by θ, are absent in multi-fidelity problems.

where $\boldsymbol{y}_{l_i}$ and $\boldsymbol{y}_h$ refer to the vectors containing the outputs of $y_{l_i}(\boldsymbol{x})$ and $y_h(\boldsymbol{x})$ at n input points (we use $n = 10^4$ throughout), and $\text{var}(\boldsymbol{y}_h)$ is the variance of $\boldsymbol{y}_h$.

3.2 Multi-Ffidelity Modeling via LMGP. Using LMGP for multi-fidelity modeling is quite straightforward. Consider the case where multiple (i.e., two or more) data sources with different levels of accuracy are available, and the goal is to emulate each source while (1) having limited data, especially from the most accurate source, (2) accounting for potential noises with unknown variance, and (3) avoiding *a priori* determination of how different sources are related to each other. The last condition indicates that we do *not* know (1) how the accuracy of the low-fidelity models compare to each other, and (2) if low-fidelity models have inherent discrepancy which may be additive or not. While not necessary, we assume it is known which data source provides the highest fidelity because this source typically corresponds to either observations/experiments or a very expensive computer model.

We assume n_h high-fidelity samples are available whose inputs and output are denoted by $\boldsymbol{x}_h$ and y_h, respectively. We also presume that the data set obtained from the ith low-fidelity source has n_{l_i} samples where the inputs and outputs are denoted via $\boldsymbol{x}_{l_i}$ and y_{l_i}, respectively.

With the above-mentioned points in mind, we use two examples in the following subsections to demonstrate our approach to multi-fidelity modeling.

3.2.1 A Simple Analytical Example. We consider a case with four data sources and use the following functions to generate data (we do not corrupt the data with noise and study the effect of noise in Sec. 4.1)

$$y_h(x) = \frac{1}{0.1x^3 + x^2 + x + 1}, \quad -2 \le x \le 3 \tag{10.1}$$

Table 1 Accuracy of data sources

	$y_{l_1}(x)$	$y_{l_2}(x)$	$y_{l_3}(x)$
RRMSE	0.23364	0.14626	0.72549

Note: RRMSEs of $y_{l_i}(x)$ are obtained using Eqs. (9) and (10).

$$y_{l_1}(x) = \frac{1}{0.2x^3 + x^2 + x + 1}, \quad -2 \le x \le 3 \tag{10.2}$$

$$y_{l_2}(x) = \frac{1}{x^2 + x + 1}, \quad -2 \le x \le 3 \tag{10.3}$$

$$y_{l_3}(x) = \frac{1}{x^2 + 1}, \quad -2 \le x \le 3 \tag{10.4}$$

where the low-fidelity sources have a nonlinear bias (compare the denominators) and are *not* ordered by the accuracy with respect to $y_h(x)$ (Table 1). Note that we do not use this knowledge of relative accuracy during multi-fidelity modeling via LMGP. Rather, by only using the datasets in LMGP, we aim to inversely discover this relation between the fidelity levels.

To perform data fusion with LMGP, we first append the inputs with one or more categorical variables that distinguish the data sources. We can use any number of multi-level categorical variables. That is, we can either (1) select a single variable with at least as many levels as there are data sources or (2) use a few multi-level categorical variables with at least as many level combinations as there are data sources. For example, with one categorical variable, we can choose $t = \{h, l_1, l_2, l_3\}$, $t = \{1, 2, 3, 4\}$, $t = \{1, a, ab, 2\}$, or $t = \{a, b, c, d, e\}$ for our pedagogical example with four data sources (in the last case level e does not correspond to any of the data sources).

For the remainder of this paper, we use two strategies for choosing categorical variables, see Fig. 2. Strategy 1 uses one categorical variable with as many levels as data sources, e.g., $t = \{a, b, c, d\}$ or $t = \{1, 2, 3, 4\}$. We add the subscript s to an LMGP that uses this strategy since a single categorical variable is used to encode the data sources. Strategy 2 employs multiple categorical variables where the number of variables and their levels equals the number of data sources[3], e.g., $t_i = \{a, b, c, d\}$ with $i = 1, 2, 3, 4$. We place the subscript m to an LMGP that uses strategy 2 to indicate that multiple categorical variables are employed. As we explain below, having more levels (or level combinations if more than one t is used) than data sources provides LMGP with more flexibility to learn the relation between the sources. This flexibility comes at the expense of having larger $\boldsymbol{A}$ and higher computational costs. As we demonstrate in Sec. 4, the performance of LMGP is relatively robust to this modeling choice as long as there are sufficient training samples and the number of latent positions does not greatly exceed the number of hyperparameters in $\boldsymbol{A}$. Regarding the latter condition, note that when LMGP must find many latent positions with a small $\boldsymbol{A}$ (i.e., a very simple map), performance may suffer due to local optimality. For example, Strategies 2 with 4 data sources results in $\Pi_{i=1}^{d_t} m_i = 4^4 = 256$ latent positions (one for each possible categorical level combination where only 4 corresponds to data sources) but there are only $d_z \times \Sigma_{i=1}^{d_t} m_i = 2 \times 16 = 32$ elements in $\boldsymbol{A}$. These elements are supposed to map the 256 points in the latent space such that the 4 points which encode the data sources have inter-distances that reflect the underlying relation between their corresponding data sources. Without sufficient data and regularization, the learned map may provide a locally optimal solution.

[3]We have tried a binary encoding version of this strategy where a data source is assigned its own categorical variable with two levels where 0 indicates the source is inactive and 1 indicates that the source is active. We found the results of this case to be similar to those of strategy 2 presented in the paper.

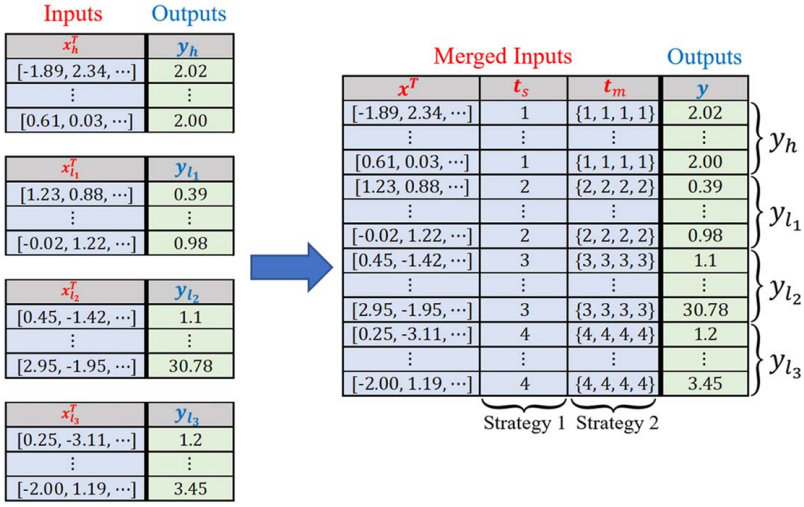

Fig. 2 Data preprocessing for multi-fidelity modeling via LMGP: We can use any number of multi-level categorical variables when fusing data with LMGP. Shown above are two strategies for choice of *t* for our example with four data sources. In strategy 1, we use one categorical variable with four levels (one for each data source) and assign each level to a unique data source. In strategy 2, we use a different categorical variable for each data source, and we give each categorical variable four levels (one for each data source) for a total of $4^4 = 256$ categorical combinations. We assign only four of these combinations to our data sources (only these four are enumerated in the figure), leaving 252 combinations unused. Note that while LMGP finds latent positions for these 252 combinations, the positions are not meaningful since they do not correspond to any of the data sources. The number of elements in the *A* matrix (see Eq. (4) that must be estimated for LMGP are 8 and 32 for the first and second strategies, respectively.

The above description clearly indicates that LMGP can, in principle, fuse any number of data sets *simultaneously*. In practice, this ability of LMGP is bounded by the natural limitations of GPs such as scalability to big data or very high dimensions. The recent advancements in GP modeling for big or high-dimensional data [38–44] have addressed these limitations to some extent and can be directly used in LMGP for multi-fidelity modeling in our future works.

For the rest of this example, we select strategy 1 and append the inputs via $t = \{1, 2, 3, 4\}$ where the number of levels equals the number of data sources. We assume the data sets are highly unbalanced and use Sobol sequence to sample from the functions in Eq. (10) with $n_h = 3$ and $n_{l_1} = n_{l_2} = n_{l_3} = 20$. Upon appending, we combine the entire data into a single training data set that is directly fed into LMGP

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_h & \mathbf{1}_{n_h \times 1} \\ \boldsymbol{X}_{l_1} & 2 \times \mathbf{1}_{n_{l_1} \times 1} \\ \boldsymbol{X}_{l_2} & 3 \times \mathbf{1}_{n_{l_1} \times 1} \\ \boldsymbol{X}_{l_3} & 4 \times \mathbf{1}_{n_{l_1} \times 1} \end{bmatrix} \text{ and } \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_h \\ \boldsymbol{y}_{l_1} \\ \boldsymbol{y}_{l_2} \\ \boldsymbol{y}_{l_3} \end{bmatrix} \tag{11}$$

where $\mathbf{1}_{n\times 1}$ is an $n \times 1$ vector of ones. The fusion results are illustrated in Fig. 3(*a*) and indicate that LMGP is able to accurately emulate each data source, including $y_h(x)$ for which only three samples are provided. As illustrated in Fig. 3(*b*), a GP fitted to only data from $y_h(x)$ provides poor performance due to lack of data.

The latent space learned by LMGP, shown in Fig. 3(*c*), provides a powerful diagnostic tool for determining correlations between data sources without prior knowledge. To understand the effect of these positions on the correlation function and hence how different data sources are related, we rewrite Eq. (3) as

$$r(\boldsymbol{u}, \boldsymbol{u}') = \exp\{-(z - z')^T (z - z')\} \cdot \exp\{-(\boldsymbol{x} - \boldsymbol{x}')^T \Omega_x (\boldsymbol{x} - \boldsymbol{x}')\} \tag{12}$$

Plugging the latent positions into Eq. (12) shows that a relative distance of $\Delta \mathbf{z}^2 = (z - z')^T (z - z')$ between two points scales the correlation function by $\exp(-\Delta \mathbf{z}^2)$. Thus, we can interpret the latent space as being a distillation of the correlations between the data sources. Note, however, that the term $\exp\{-(\boldsymbol{x} - \boldsymbol{x}')^T \boldsymbol{\Omega}_x (\boldsymbol{x} - \boldsymbol{x}')\}$, which accounts for the correlation between outputs at different points in the input space, remains the same as we change data sources. Thus, our modeling assumption is that this correlation is similar for all data sources. In layman's terms, we expect each data source to have a relatively similar shape. This is often true in multi-fidelity problems and if this modeling assumption is not met, LMGP estimates Ω_x to provide the best compromise between different sources, which may provide poor performance in emulation for some or all sources. To avoid making such a compromise, we can use the latent space to identify the dissimilar data source(s) and then repeat the fusion process after excluding them.

Note also that the objective function in Eq. (8) that is used to find the latent positions is invariant under translation and rotation. In order to find a unique solution, we enforce the following constraints in two dimensions (more constraints are needed for $d_z > 2$): latent point 1 is placed at the origin, latent point 2 is positioned on the positive x axis, and latent point 3 is restricted to the $y > 0$ half-plane. We assign $y_h(x)$ to position 1 for both of our strategies as it yields more readable latent plots, but this choice is arbitrary and does not affect the relative distances between the latent positions as shown in Sec. 4.

Returning to our example with the above constraints in mind, we can see that the latent points corresponding to $y_h(x)$ and $y_{l_2}(x)$ are close and the other points relatively distant, especially the point representing $y_{l_3}(x)$. This observation matches with our knowledge of the relative accuracies of the underlying functions with respect to $y_h(x)$ (this knowledge is *not* provided to LMGP). In other words, LMGP has accurately determined the correlations between the data sources despite the sparse sampling for $y_h(x)$. Given that $y_{l_2}(x)$ appears to be much more accurate than other low-fidelity sources with respect to $y_h(x)$, one might consider fitting LMGP using only data from these two sources rather than all of the data to produce a more accurate high-fidelity emulator. The results of this approach, shown in Fig. 3(*d*), demonstrate that high-fidelity emulation performance is actually equivalent with all sources

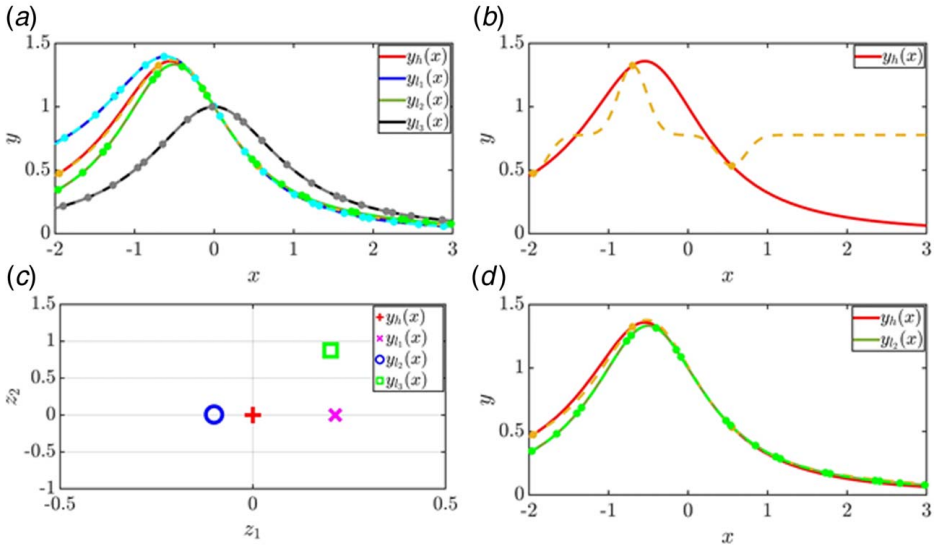

Fig. 3 Approaches to data fusion: (*a*) LMGP with all available data: LMGP fit to all available data is able to emulate each data source with high accuracy. The inaccuracy of y_{l_3} does not negatively impact high-fidelity emulation performance. (*b*) Standard GP: Standard GP fit to only the three available high-fidelity samples performs poorly. (c) Learned latent space: LMGP only uses four data sets to learn a latent space that indicates how "close" different data sources are with respect to each other. While the data sets are quite unbalanced ($n_h = 3$ and $n_{l_1} = n_{l_2} = n_{l_3} = 20$), LMGP can clearly visualize the relative accuracy of each low-fidelity model with respect to the high-fidelity data. (*d*) LMGP with only $y_{l_2}(x)$ and $y_h(x)$: Despite the fact that $y_{l_2}(x)$ misrepresents $y_h(x)$ in some regions, LMGP is able to use correlations between the two sources to accurately emulate $y_h(x)$ with approximately equivalent accuracy to when all sources are used.

used; i.e., using less accurate sources does not make our estimate of $y_h(x)$ worse in this case because they include useful information about $y_h(x)$.

We can also explicitly compare the correlations found by LMGP to other similarity metrics, such as cosine similarity, S_c

$$S_C(y_h(x),\ y_{l_i}(x)) = \frac{\boldsymbol{y}_h^T \cdot \boldsymbol{y}_{l_i}}{\|\boldsymbol{y}_h\| \cdot \|\boldsymbol{y}_{l_i}\|} \tag{13}$$

which we calculate using the same 10,000-point vectors as RRMSE. The rough latent distances between the point corresponding to $y_h(x)$ and the points corresponding to $y_{l_1}(x)$, $y_{l_2}(x)$, and $y_{l_3}(x)$ are, respectively, 0.21, 0.10, and 0.90, which correspond to correlations of, respectively, 0.96, 0.99, and 0.45 using Eq. (12). By contrast, the rough cosine similarities are, respectively, 0.994, 0.997, and 0.911. While both measures show the same relative relationships between data sources in terms of which source has the most/least correlation/similarity, LMGP finds a much starker difference between $y_{l_3}(x)$ and $y_h(x)$ than the cosine similarity metric. The correlations found by LMGP better match both the RRMSE measures and the intuitive relative similarity of the functions based on looking at their plots. Note that while the cosine similarity is calculated using 10,000 test points from the analytic functions, LMGP calculates its correlation measurements based purely on the training data, i.e., three samples from the high-fidelity function and 20 samples from each low-fidelity function.

In order to support our assertion that a two-dimensional latent space is typically sufficient to encode the relationships between data sources, we show the latent space for LMGP fits all data sources with $d_z = 3$ in Fig. 4. We enforce the following constraints in three dimensions: latent point 1 is placed at the origin, latent point 2 is positioned on the positive z_1 axis, latent point 3 is restricted to the $z_3 = 0$ & $z_2 \geq 0$ half-plane, and latent point 4 is restricted to $z_3 \geq 0$. These constraints reduce degrees-of-freedom by restricting translation, rotation, and reflection. In this case, we find that the relative distances between the latent points in Fig. 4 are nearly the same as those in Fig. 3(*c*), which indicates that two dimensions are sufficient to encode the relationships between the data sources.

3.2.2 Effect of Categorical Variable Assignment. We now consider an example with three datasets drawn from the following functions:

$$y_h(x) = 0.1x^3 + x^2 + x + 1, \quad -2 \leq x \leq 3 \tag{14.1}$$

$$y_{l_1}(x) = 0.2x^3 + x^2 + x + 1, \quad -2 \leq x \leq 3 \tag{14.2}$$

$$y_{l_2}(x) = x^2 + x + 1, \quad -2 \leq x \leq 3 \tag{14.3}$$

where we again sample via Sobol sequence with $n_h = 3$, $n_{l_1} = n_{l_2} = 20$, and do not apply noise to the samples. We create 30 unique quasi-random iterations (hereafter referred to as repetitions) to examine the robustness of our approach to sampling variations. As shown in Fig. 5(*a*), both $y_{l_2}(x)$ and $y_{l_1}(x)$ are equally accurate as they differ from $y_h(x)$ by a $\pm 0.1x^3$ term. This time, we fit LMGP using both strategies for categorical variable assignment and examine the effect of this choice as well as the size of the training data sets on the results. We use the subscript *All* to denote the

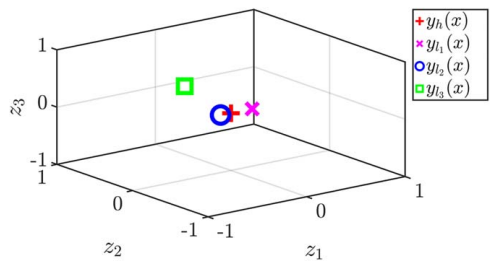

Fig. 4 Learned latent space with $d_z = 3$: LMGP finds the latent positions to lie on a two-dimensional subspace

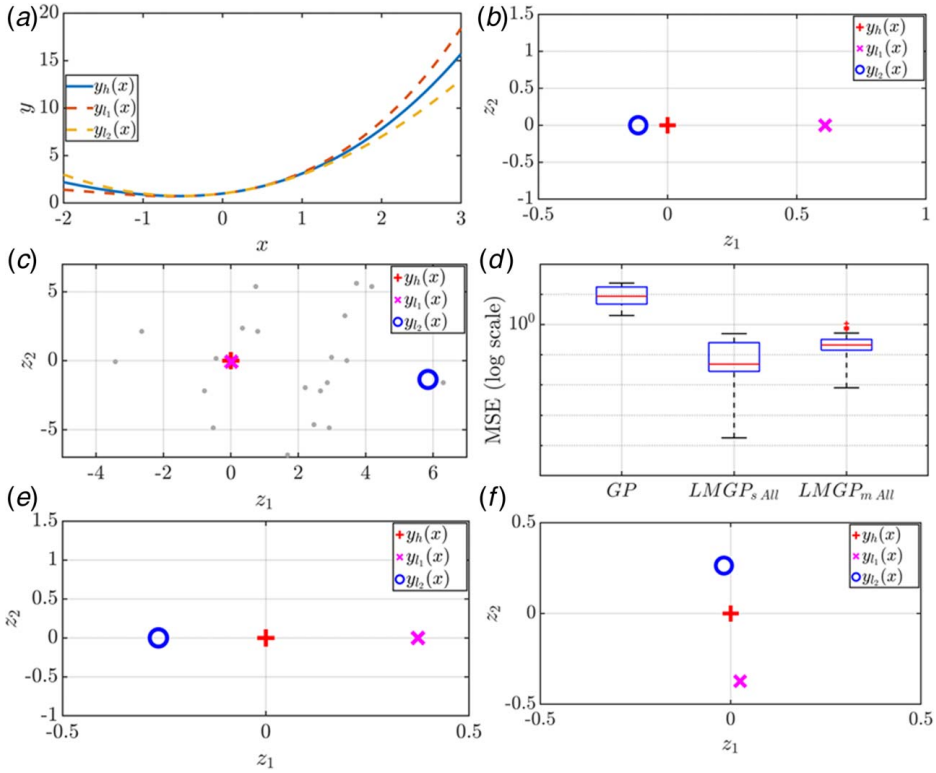

Fig. 5 Approaches to categorical variable assignment: (*a*) accuracy of data sources: Both low-fidelity sources are equally accurate, i.e., they have the same RRMSE with respect to $y_h(x)$. (*b*) Latent space for $LMGP_{s\ All}$: We show the latent space for one repetition, but LMGP consistently finds one source to be close to and another to be distant from the position for $y_h(x)$ across repetitions. (*c*) Latent space for $LMGP_{m\ All}$: We show the latent space for one repetition. The positions and relative distances are not consistent across repetitions. The gray dots correspond to latent positions that do not correspond to any data source. (*d*) High-fidelity emulation performance across 30 repetitions: LMGP outperforms GP in high-fidelity emulation for both categorical variable strategies. MSEs are calculated by comparing emulator predictions to analytic function outputs at 10,000 points. (*e*) and (*f*) Latent spaces with more data: With more data, $LMGP_{s\ All}$, shown in (*e*), and $LMGP_{m\ All}$, shown in (*f*), consistently find latent positions that accurately reflect the relative accuracies of the data sources. We do not show the latent positions not corresponding to any data sources in (*f*), and as such, the shown points do not conform to the 2D constraints.

fact that we fit LMGP to all available data and employ the subscripts l_i to refer to an LMGP fitted via only $\boldsymbol{y}_h$ and $\boldsymbol{y}_{l_i}$.

The latent space for LMGP using one categorical variable is demonstrated in Fig. 5(*b*) and shows that this strategy enables LMGP to learn that both sources have inaccuracy with respect to $y_h(x)$. However, LMGP consistently finds one source to be significantly more accurate than the other as a result of the sparse sampling. By contrast, the positions found by LMGP using multiple categorical variables are very inconsistent across repetitions and often estimate one of the sources as being either extremely correlated or uncorrelated with $y_h(x)$ (Fig. 5(*c*)). This inconsistency is because $LMGP_{m\ All}$ has quite a few hyperparameters (1 roughness parameter and 18 parameters in the $\boldsymbol{A}$ matrix), which are difficult to estimate with scarce data. Across the repetitions of $LMGP_{m\ All}$, at least one data source is always found to be well correlated with $y_h(x)$ so high-fidelity predictions are still good and much better than fitting a traditional GP to only the high-fidelity data (Fig. 5(*d*)). When we increase the available data to $n_h = 15$, $n_{l_1} = n_{l_2} = 50$, both $LMGP_{s\ All}$ and $LMGP_{m\ All}$ consistently (i.e., across repetitions) find latent positions for the low-fidelity functions that are approximately equidistant from $y_h(x)$. We demonstrate this in Fig. 6, which shows histograms of the distances between the latent points for $y_h(x)$ and $y_{l_1}(x)$ or $y_{l_2}(x)$ in (*a*) and (*b*), respectively. Notably, $LMGP_{m\ All}$ is less consistent in both cases, with a few poor-performing outliers in Fig. 6(*b*). Interestingly, the positions for the two low-fidelity sources are in opposite directions from $y_h(x)$ which agrees with the fact that discrepancies are equal but of opposite sign (Figs. 5(*e*) and 5(*f*)). Notably, as we show in Fig. 7, this property is not a result of the constraints we apply to the latent points during fitting and persists even when no constraints are applied; i.e., all three points lie on a line.

While we did not apply noise to the samples in these pedagogical examples, as we demonstrate in Sec. 4, LMGP is fairly robust to noise both with respect to emulation performance and finding latent positions.

3.3 Calibration via LMGP. Calibration problems closely resemble multi-fidelity modeling in that a number of high- and low-fidelity data sets are assimilated or fused together. However, in such problems, low-fidelity data sets[4] typically involve calibration inputs which are not directly controlled, observed, or measured in the high-fidelity data (i.e., high-fidelity data have fewer inputs). Hence, in addition to building surrogate models, one seeks to *inversely* estimate these inputs during the calibration process.

Following previous sections, we denote the quantitative and latent representation of the qualitative inputs via $\boldsymbol{x}$ and $\boldsymbol{z}$,

[4]Generally built via computer simulations.

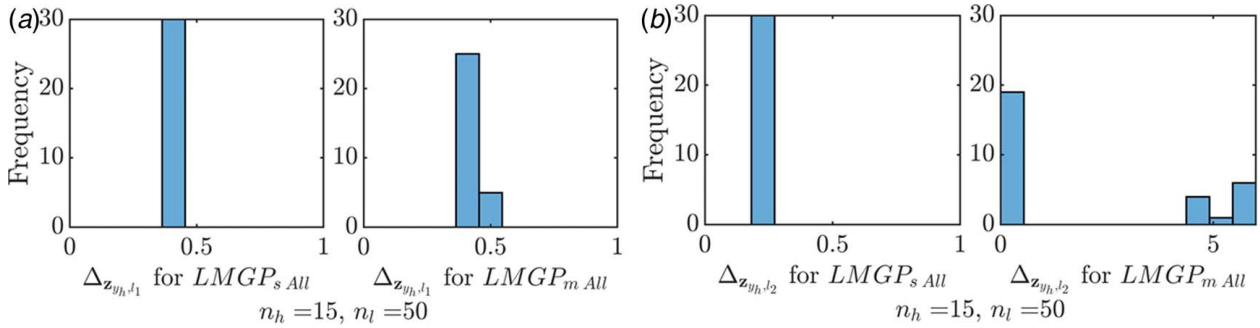

Fig. 6 Histogram of latent distances: The above figure shows a histogram of the relative distances across 30 repetitions between y_h and each low-fidelity source for both strategies. (*a*) y_h and y_{l_1}: Both strategies find similar distances, with $\text{LMGP}_{m\ All}$ being only slightly less consistent. (*b*) y_h and y_{l_2}: Both strategies again find similar distances. This time, however, $\text{LMGP}_{m\ All}$ displays a higher number of poor-performing outliers.

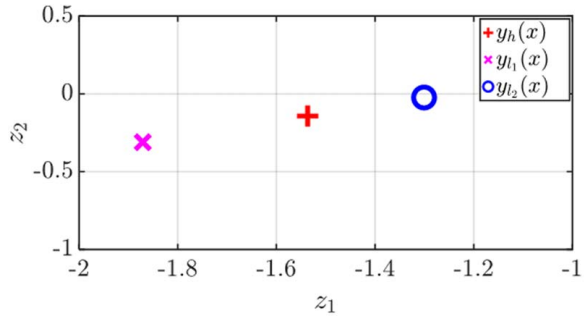

Fig. 7 Latent space with no constraints: The relative relationships between the data sources in the latent space remain the same without applying constraints to the locations of the points

respectively (note that z encodes data sources as per Sec. 3.2). While these inputs are shared across all data sources, the low-fidelity data sources have additional quantitative inputs, $\boldsymbol{\theta}$, whose "best" values must be estimated using the high-fidelity data. We represent these "best" values by $\boldsymbol{\theta}^*$ which minimize the discrepancies between low- and high-fidelity data sets based on an appropriate metric. In the case that one wishes to calibrate and assimilate multiple computer models simultaneously, we assume that the calibration parameters are shared between the low-fidelity data sets and are expected to have the same best value. Our estimate of $\boldsymbol{\theta}^*$ is denoted by $\hat{\boldsymbol{\theta}}$ and is obtained via MLE by modifying LMGP's correlation function as

$$r\left(\begin{bmatrix}\boldsymbol{x}\\ z\\ \boldsymbol{\theta}\end{bmatrix}^{(i)}, \begin{bmatrix}\boldsymbol{x}\\ z\\ \boldsymbol{\theta}\end{bmatrix}^{(j)}\right) = \exp\{-(z^{(i)}-z^{(j)})^T(z^{(i)}-z^{(j)})\} \times \exp\{-(\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)})^T\Omega_{\boldsymbol{x}}(\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)})\} \times \exp\{-(\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^{(j)})^T\Omega_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^{(j)})\} \tag{15}$$

where $\boldsymbol{x}^{(i)}$, $\boldsymbol{x}^{(j)}$, $\boldsymbol{\Omega}_{\boldsymbol{x}}$, $z^{(i)}$, and $z^{(j)}$ are defined as before. $\boldsymbol{\theta}^{(i)}$ denotes the calibration parameters of sample i and $\Omega_{\boldsymbol{\theta}}$ is the diagonal matrix of roughness/scale parameters associated with $\boldsymbol{\theta}$. When one or both of the inputs to the correlation function lack calibration parameters (i.e., at least one of the inputs corresponds to a high-fidelity sample), we substitute $\hat{\boldsymbol{\theta}}$ in the last term of Eq. (15). If both inputs are from the high-fidelity data, the term $\exp\{-(\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^{(j)})^T\boldsymbol{\Omega}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^{(j)})\}$ does not affect the correlation because $\exp\{-(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}})^T\boldsymbol{\Omega}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}})\} = \exp\{0\} = 1$

Using Eq. (15), we see that in a calibration problem with multiple data sources all the hyperparameters of an LMGP can be estimated by MLE in the same way that a traditional GP is trained, i.e., by optimizing the following objective function where the correlation matrix is built via Eq. (15)

$$[\hat{\boldsymbol{\omega}},\ \hat{\boldsymbol{A}},\ \hat{\boldsymbol{\theta}},\ \hat{\boldsymbol{\Omega}}_{\boldsymbol{\theta}}] = \underset{\boldsymbol{\omega},\boldsymbol{A},\boldsymbol{\theta},bi\Omega_{\boldsymbol{\theta}}}{\operatorname{argmin}}\ n\log(\hat{\sigma}^2) + \log(|\boldsymbol{R}|) = \underset{\boldsymbol{\omega},\boldsymbol{A},\boldsymbol{\theta},\boldsymbol{\Omega}_{\boldsymbol{\theta}}}{\operatorname{argmin}}\ L, \tag{16}$$

Preprocessing the data for calibration via LMGP is schematically illustrated in Fig. 8. Following the same procedure described in Sec. 3.2, we append the inputs with categorical variables to distinguish data sources. We also augment the high-fidelity inputs with some unknown values to account for the missing calibration parameters. Once the mixed data set that contains *all* the low- and high-fidelity data are built, we directly use it in LMGP to not only build emulators for each data source but also estimate $\hat{\boldsymbol{\theta}}$. Similar to multi-fidelity modeling, any number of data sets can be simultaneously used via LMGP for calibration.

We now illustrate the capabilities of LMGPs for calibration via two analytical examples where there are one high-fidelity data source $y_h(x)$ and up to two low-fidelity data sources, denoted by $y_{l_1}(x)$ and $y_{l_2}(x)$. We presume that in both examples the goals are to accurately emulate the high-fidelity data source and estimate the calibration parameters. We note that once an LMGP is trained, it provides an emulator for each data source but here we only evaluate accuracy for surrogating $y_h(x)$ since much fewer data points are available from it, and hence, emulating it is more difficult.

3.3.1 A Simple Calibration Problem. For our first example, we consider the polynomials in Eq. (17) as data sources and take five samples from $y_h(x)$ and 25 samples from each of $y_{l_1}(x)$ and $y_{l_2}(x)$ (none of the datasets are corrupted with noise)

$$y_h(x) = 0.1x^3 + x^2 + x + 1, \quad -2 \le x \le 3 \tag{17.1}$$

$$y_{l_1}(x) = \theta x^3 + x^2 + x + 1, \quad -2 \le x \le 3, \quad -2 \le \theta \le 2 \tag{17.2}$$

$$y_{l_2}(x) = \theta x^3 + x^2 + 1, \quad -2 \le x \le 3, \quad -2 \le \theta \le 2 \tag{17.3}$$

We set θ^* as 0.1 because it is the true value of the coefficient on the leading x^3 term. Note that $y_{l_1}(x)$ can match $y_h(x)$ perfectly with an appropriate choice of θ; i.e., $y_{l_1}(x)$ has no model form error when $\hat{\theta} = 0.1$ (Fig. 9(*a*)). Conversely, no value of θ allows $y_{l_2}(x)$ to match $y_h(x)$ since $y_{l_2}(x)$ has a linear model form error. When solving this calibration problem, we assume there is no knowledge on whether low-fidelity models have discrepancies and expect the learned latent space of LMGP to provide diagnostic measures that indicate potential model form errors.

As shown in Fig. 9(*b*), the learned latent positions by LMGP are quite consistent with our expectations despite the fact that limited and unbalanced data are used in LMGP's training. It is evident

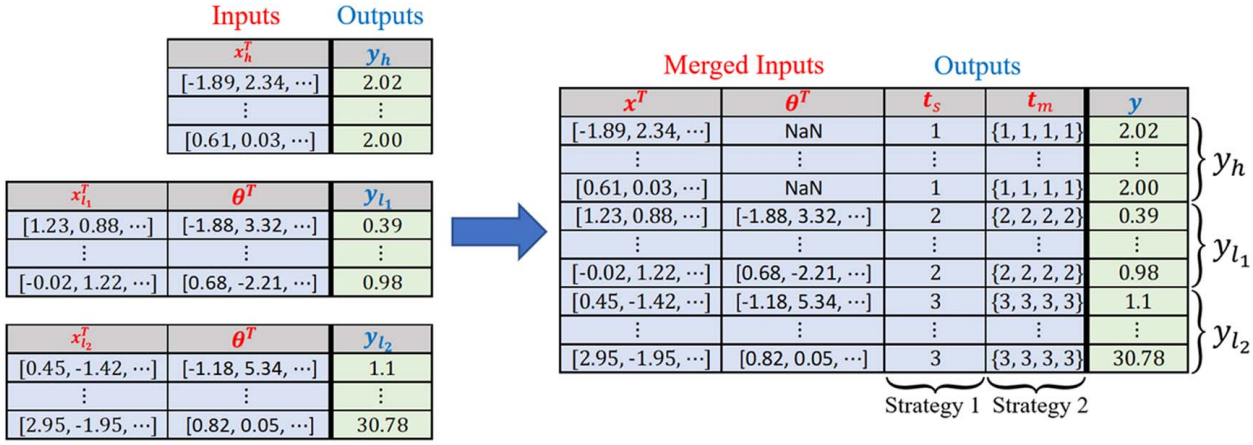

Fig. 8 Preprocessing of data for calibration: Multiple data sets are combined in a specific way and then directly used by LMGP. The high-fidelity data are augmented with NaNs since they lack calibration parameters, and all data are augmented with categorical IDs that denote the source from which a datum is drawn. We use strategy 1 for choice of *t* in both examples in Sec. 3.3.

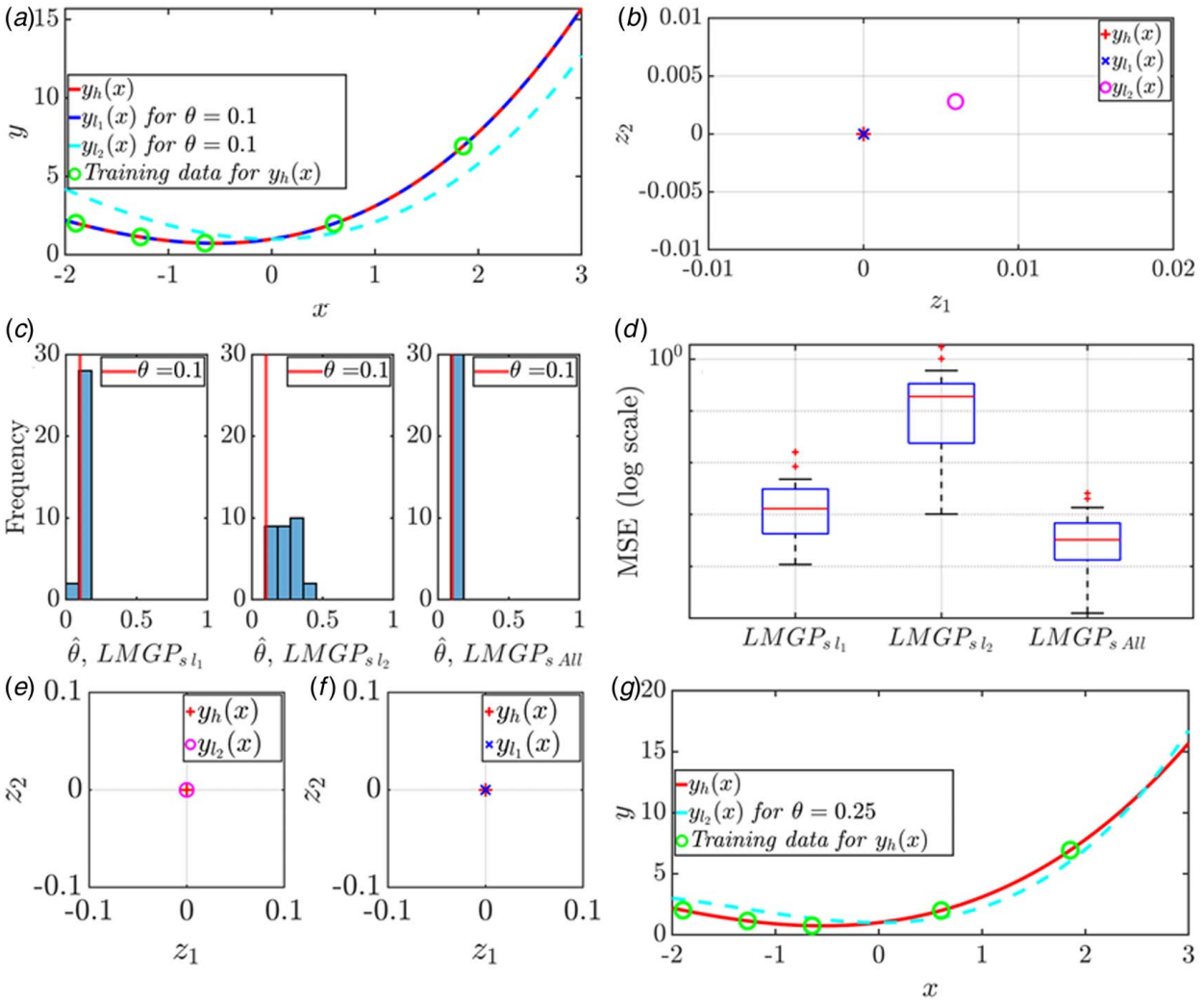

Fig. 9 Calibration with LMGP: (*a*) Underlying functions with true calibration parameters: $y_{l_1}(x)$ and $y_h(x)$ are coincident for $\theta = \theta^*$. (*b*) Latent space for $LMGP_{s\,All}$: Latent positions for $y_h(x)$ and $y_{l_1}(x)$ are coincident while the position for $y_{l_2}(x)$ is relatively more distant (albeit still quite close). (c) Histogram of estimated calibration parameters: We estimate θ over 30 repetitions where the LMGP fitted via all data yields more consistent estimates. All three models use a single categorical variable to encode data sources. (*d*) High-fidelity emulation performance: Using all data yields the best performance since data sources are correlated. (e) and (*f*) Latent space for $LMGP_{s\,l_2}$ and $LMGP_{s\,l_1}$: LMGP cannot detect model form error between $y_h(x)$ and $y_{l_2}(x)$ since data are scarce and an appropriately estimated θ enables $y_{l_2}(x)$ to resemble $y_h(x)$ fairly well as shown in (e). LMGP can correctly detect that $y_{l_1}(x)$ does not have model form error, as shown in (*f*). (*g*) $y_{l_2}(x)$ with estimated calibration parameters versus $y_h(x)$: $y_{l_2}(x)$ can nearly interpolate sparse training data for $y_h(x)$ with the appropriate calibration parameter.

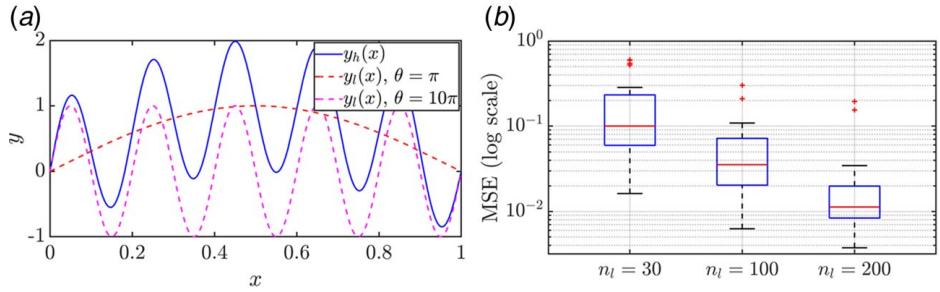

Fig. 10 Calibration via LMGP: (*a*) Plot of the underlying functions: Due to model form error, y_l(x) is unable to capture the behavior of y_h(x) regardless of the choice of θ. Choosing $\theta = \pi$ indicates a discrepancy of sin(10π), while choosing $\theta = 10\pi$ indicates a discrepancy of sin(π). Notably, the analytic MSEs (calculated by comparing y_h(x) and y_l(x) at 10,000 sample points equally spaced over the input range) for both choices of theta are 0.5, i.e., the magnitude of the error is the same for both choices of θ. (*b*) High-fidelity emulation performance: As we provide more low-fidelity data, LMGP's performance on high-fidelity emulation increases.

that the latent positions corresponding to $y_h(x)$ and $y_{l_1}(x)$ are very close to each other, indicating negligible model form error. In contrast, the positions corresponding to $y_h(x)$ and $y_{l_2}(x)$ are more distant which signals that $y_{l_2}(x)$ has model form error.

The learned latent positions in Fig. 9(*b*) suggest that $y_{l_1}(x)$ (when calibrated properly) captures the behavior of $y_h(x)$ better than $y_{l_2}(x)$. Correspondingly, one may argue calibrating $y_{l_1}(x)$ individually may improve performance. To assess this argument, we fit LMGPs to three combinations of the available data sets and compare the performance of these LMGPs in terms of estimating θ^* and emulating $y_h(x)$. In all three cases, we use a single categorical variable to encode the data source, and hence, the subscript s is appended to the model names (so, $\text{LMGP}_{s\,l_1}$ calibrates $y_{l_1}(x)$ via $y_h(x)$ and uses a single categorical variable). The results are shown in Figs. 9(*c*) and 9(*d*) and indicate that using both low-fidelity data sets provides the best performance since (1) $\hat{\theta}$s are estimated more consistently as the distribution is centered at θ^* with small variations, and (2) errors (measured in terms of mean squared error, MSE) for predicting $y_h(x)$ are smaller. These observations can be explained by the fact that the highest relative distance between data sources in Fig. 9(*b*) is on the order of 0.05, which indicates that LMGP finds $y_{l_2}(x)$ to be very similar to $y_h(x)$ and $y_{l_1}(x)$ as this distance scales the correlation function by $\exp\{(-0.05)^2\} \approx 0.998$. That is, LMGP can distill useful knowledge from the correlation between y_{l_2} and other sources to improve its performance in estimating θ and emulating $y_h(x)$. When $y_{l_1}(x)$ is excluded from the calibration process and only $y_{l_2}(x)$ is used in calibration, LMGP provides biased and less consistent estimates for θ and relatively large MSEs for predicting $y_h(x)$.

While the distance in the latent space typically encodes model form error that is not reduceable by adjusting θ, LMGP may mistake model form error for noise in the case that certain calibration parameters allow the low-fidelity model to closely match the high-fidelity function. This is the case if we fit LMGP to only $y_h(x)$ and $y_{l_2}(x)$. As shown in Fig. 9(*e*), LMGP places the latent positions for $y_h(x)$ and $y_{l_2}(x)$ very close to each other when $y_{l_1}(x)$ is excluded. We explain this observation by referring back to Fig. 9(*c*) where $\text{LMGP}_{s\,l_2}$ finds $\hat{\theta} \approx 0.25$. Plotting $y_{l_2}(x)$ for this value of θ reveals that it can nearly interpolate the training data (Fig. 9(*g*)). As such, LMGP mistakes 0.25 for the true value of θ and dismisses the small resultant error as noise. This also explains the aforementioned bias and inconsistency in estimating θ across repetitions as the value that comes closest to interpolating $y_h(x)$ is different depending on sampling variations. By contrast, LMGP fit to all data is able to leverage the information from $y_{l_1}(x)$ to determine that $y_{l_2}(x)$ has model form error. And, as expected, no model form error is indicated in the latent space if only $y_{l_1}(x)$ is used in calibration (Fig. 9(*f*)).

As this simple example clearly indicates, a simultaneous fusion of *multiple* (i.e., more than 2) data sources can decrease identifiability issues in calibration. This property is one of the main strengths of our data fusion approach.

3.3.2 Calibration With Severe Model Form Error. In our second analytical example, we examine a case where there is only one low-fidelity source which has a significant model form error

$$y_h(x) = \sin(\pi x) + \sin(10\pi x), \quad 0 \le x \le 1 \tag{18.1}$$

$$y_l(x) = \sin(\theta x), \quad 0 \le x \le 1 \text{ and } \pi - 2 \le \theta \le 10\pi + 2 \tag{18.2}$$

Based on Eq. (18), θ^* can be either π or 10π so the range of θ in $y_l(x)$ is chosen wide enough to include both values. As shown in Fig. 10(*a*), considering $\theta^*=\pi$ implies that the high-fidelity source is either noisy or has a high-frequency component that is missing from the low-fidelity source (note that in realistic applications the functional form of data sources is unknown so high-frequency trends can be easily misclassified as noise in which case they are typically smoothed out, i.e., not learned). Conversely, considering $\theta^*=10\pi$ implies that $y_l(x)$ is expected to surrogate the high-frequency component of $y_h(x)$ and that $\sin(\pi x)$ is the discrepancy. Note that the analytic MSEs (calculated by comparing $y_h(x)$ and $y_l(x)$ at 10,000 sample points equally spaced over the input range) and cosine similarities (between $y_h(x)$ and $y_l(x)$, also at 10,000 sample points equally spaced over the input range) are identical for each choice of θ, i.e., both choices yield a discrepancy of the same magnitude, and we cannot determine which choice is better *a priori* based on MSEs or cosine similarity. We are interested in finding out which value is a better estimate for θ^* and whether LMGP is able to consistently infer this value purely from the low- and high-fidelity data sets. We do not corrupt the data sets with noise and investigate the effect of noise in Sec. 4.2.

We now explore the effects of the low-fidelity data set size on the performance while holding the number of high-fidelity data constant. Specifically, we examine $n_l = 30$, 100, 200 with $n_h = 15$ in each case. Note that standard GP trained on only the 15 available high-fidelity samples cannot learn the high-frequency behavior of $y_h(x)$ and instead interprets it as noise.

As shown in Fig. 10(*b*), increasing n_l improves high-fidelity prediction and we can therefore consider the estimates of θ and the latent distances in the $n_l = 200$ case to be the most accurate since they maximize prediction performance. Shown in Fig. 11(*a*) are histograms of the latent distances over 30 repetitions for each case. When few low-fidelity data are available, the latent distances are close to zero; with plentiful data, the latent distances are clustered around 0.5. This indicates that LMGP interprets $y_h(x)$ and $y_l(x)$ as

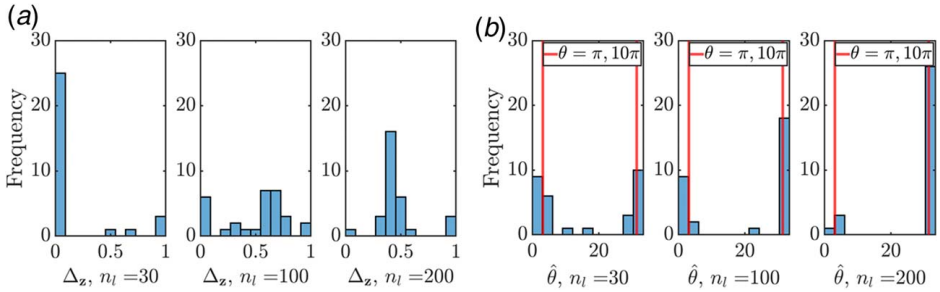

Fig. 11 Analysis for sin wave example: (*a*) Histogram of latent distances: LMGP estimates distances near zero and 0.5 with a few and plentiful data points, respectively. There is a large variance in the latent distances for $n_l = 100$, with a large spike at zero and a cluster near 0.5 which correspond to LMGP's estimates for $n_l = 30$ and $n_l = 200$ respectively. That is, as the size of the data is increasing, LMGPs interpretation of model form error changes. (*b*) Histogram of $\hat{\theta}$: As more low-fidelity data are provided, estimates become more closely clustered around 10π. With few low-fidelity data, LMGP guesses $\theta = \pi$ almost half of the time but with $n_l = 200$ LMGP almost consistently guesses $\theta = 10\pi$ which means that $y_l(x)$ has a high-frequency behavior.

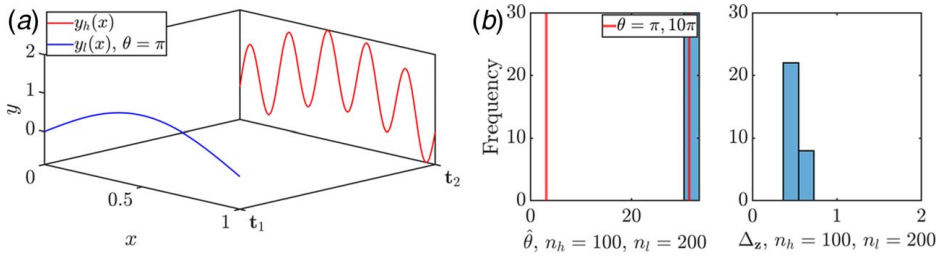

Fig. 12 Effect of categorical variable and data set size: (*a*) Effect of shifting the level: With $\hat{\theta} = \pi$ the shift in the categorical variable is supposed to "model" sin(10πx), which is much more difficult than the alternative. (*b*) Effect of data set size: with $n_h = 100$ and $n_l = 200$ LMGP consistently estimates θ as 10π so the shift in categorical variable learns the simplest discrepancy candidate, i.e., sin(πx).

being closely correlated when we have few low-fidelity data, but consistently learns that $y_l(x)$ has a noticeable error with respect to $y_h(x)$ as we provide more data. Without sufficient low-fidelity data, LMGP learns the low-frequency behavior of $y_h(x)$ which follows $\sin(\pi x)$ and dismisses the high-frequency behavior as noise. Consequently, LMGP finds a small latent distance since $y_l(x)$ can capture $\sin(\pi x)$ without error.

We now examine the histogram of $\hat{\boldsymbol{\theta}}$ in Fig. 11(*b*). When few low-fidelity data are available, estimates are clustered around both π and 10π while with plentiful data the estimates are tightly clustered around only 10π. This observation indicates that when little data are available, LMGP interprets $y_h(x)$ to more closely resemble $\sin(\pi x)$ almost half of the times which matches with the observation on the learned latent distances; i.e., the high-frequency behavior is interpreted as noise and not learned. As more low-fidelity data are available, LMGP is able to learn the high-frequency behavior of $y_h(x)$ using the low-fidelity data and interprets $y_h(x)$ as more closely resembling $\sin(10\pi x)$.

Why does LMGP prefer $\hat{\theta} = 10\pi$ with more data? To answer this question, we note that in LMGP shifting the levels of the categorical variable is expected to reflect a change in data source. With $\hat{\theta} = \pi$, the shift in the categorical variable is supposed to "model" $\sin(10\pi x)$, which is much more difficult than the alternative. In other words, LMGP is trying to learn the simplest function that must be represented by a shift in the categorical variable (Fig. 12(*a*)). We further explore this conjecture by fitting an LMGP to 100 noiseless samples from $y_h(x)$ and 200 samples from $y_l(x)$. This amount of data is sufficient to learn both the high-frequency behavior of $y_h(x)$ and the high-frequencies of $y_l(x)$ (i.e., the behavior of $y_l(x)$ for large θ), and as such, we expect the latent positions and calibration estimates found by LMGP in this case to be optimal. As shown in Fig. 12(*b*), LMGP finds latent distances near 0.5 and $\theta = 10\pi$ very consistently; i.e., LMGP prefers to estimate the calibration parameters to minimize the complexity of the discrepancy function.

4 Results

To validate our approach in both multi-fidelity and calibration problems, we test our method on analytical functions and assess its performance against competing methods. In each example, we vary the size of the training data and the added noise variance and repeat the training process to account for randomness (20 times for the multi-fidelity problems and 30 times for the calibration problems). The knowledge of the value of the noise variance is *not* used in training. To measure accuracy, we use 10, 000 noisy test samples to obtain MSE (note that since the test data are noisy, the MSE obtained by an emulator cannot be smaller than the noise variance).

In our LMGP implementation, we always use $d_z = 2$ and select $-3 \le a_{i,j} \le 3$ during optimization where $a_{i,j}$ are the elements of the mapping matrix $\boldsymbol{A}$. When using LMGP for calibration, the search space for each element of $\hat{\boldsymbol{\theta}}$ is restricted to $[-2, 3]$ after scaling the data to the range $[0, 1]$ (i.e., we select a search space larger than the sampling range for $\boldsymbol{\theta}$). We use the modular version of KOH's approach where we set a uniform prior for $\boldsymbol{\theta}$ over the sampling range defined in each problem statement. All optimizations are done based on the L-BFGS method, which is a second-order gradient-based optimization technique.

4.1 Multi-Fidelity Results. We consider two analytical problems with high-dimensional inputs. In the first multi-fidelity problem, we consider a set of four functions that model the weight of a light aircraft wing [45]

Table 2 Relative accuracy of functions for wing-weight problem

	$y_{l_1}(x)$	$y_{l_2}(x)$	$y_{l_3}(x)$
RRMSE	0.19912	1.1423	5.7484

Note: The functions are listed in decreasing order with respect to accuracy, with $y_{l_3}(\boldsymbol{x})$ being especially inaccurate. 10000 points are used in calculating RRMSE.

$$y_h(\boldsymbol{x}) = 0.036 S_\omega^{0.758} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_\omega W_p \tag{19.1}$$

$$y_{l_1}(\boldsymbol{x}) = 0.036 S_\omega^{0.758} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + 1 \times W_p \tag{19.2}$$

$$y_{l_2}(\boldsymbol{x}) = 0.036 S_\omega^{0.8} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + 1 \times W_p \tag{19.3}$$

$$y_{l_3}(\boldsymbol{x}) = 0.036 S_\omega^{0.9} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100 t_c}{\cos(\Lambda)}\right)^{-0.3} (N_z W_{dg})^{0.49} + 0 \times W_p \tag{19.4}$$

$$\boldsymbol{x}^T = [S_\omega,\ W_{f\omega},\ A,\ \Lambda,\ q,\ \lambda,\ t_c,\ N_z,\ W_{dg},\ W_p]$$

$$\min(\boldsymbol{x}) = [150,\ 220,\ 6,\ -10,\ 16,\ 0.5,\ 0.08,\ 2.5,\ 1700,\ 0.025]$$

$$\max(\boldsymbol{x}) = [200,\ 300,\ 10,\ 10,\ 45,\ 1,\ 0.18,\ 6,\ 2500,\ 0.08]$$

These functions are ten-dimensional and have varying degrees of fidelity where, following the notation introduced in Sec. 3, $y_h(\boldsymbol{x})$ has the highest fidelity. Note that in $y_{l_3}(\boldsymbol{x})$ we multiply W_p by zero which is equivalent to reducing the dimensionality of the function by one. As enumerated in Table 2, the above functions are listed in decreasing order with respect to accuracy; that is, $y_{l_1}(\boldsymbol{x})$ and $y_{l_3}(\boldsymbol{x})$ are the most and least accurate models, respectively. Table 2 is generated by evaluating the four functions in Eq. (19) on the same 10, 000 inputs as described in Sec. 3.2 (no noise is added to the outputs). This knowledge of relative accuracy of the data sources is *not* used when fitting an LMGP.

We consider various amounts of available low-fidelity data, with and without noise. We also compare the two different settings introduced in Sec. 3.2 where subscripts s and m indicate whether a single or multiple categorical variables are used to encode the data sources in LMGP. We only take 15 samples for $y_h(\boldsymbol{x})$, which is a very small number given the high dimensionality of the input space. Additionally, we investigate the effect of fusing the four datasets jointly against fusing the high-fidelity data with each of the low-fidelity sources (in the former case the subscript *All* is appended to LMGP while in the latter case l_1, l_2 or l_3 is used in the subscript depending on which source is used in addition to $y_h(\boldsymbol{x})$).

The results are summarized in Fig. 13 and indicate that the different versions of LMGPs consistently outperform traditional GPs (only fitted to high-fidelity data) in all cases, even when only using the least accurate data source to augment high-fidelity emulation. This superior performance of LMGP is due to taking advantage of the correlations between datasets that compensates, to some extent, for the sparsity of the high-fidelity data. LMGP also has the advantage in consistency where fewer outliers are observed in MSE compared to GP. This consistency indicates that our modeling assumptions (e.g., how to encode the data source) marginally affect the performance in this example.

In cases without noise, i.e., Figs. 13(*a*) and 13(*c*), LMGPs fit to the data from $y_{l_1}(\boldsymbol{x})$ and $y_h(\boldsymbol{x})$ perform on par with or better than the LMGPs that are fit to all data and the small differences are mostly due to sample-to-sample variations. However, in cases with noise, i.e., Figs. 13(*b*) and 13(*d*), using all the data sets improves the performance of LMGP. We explain this observation

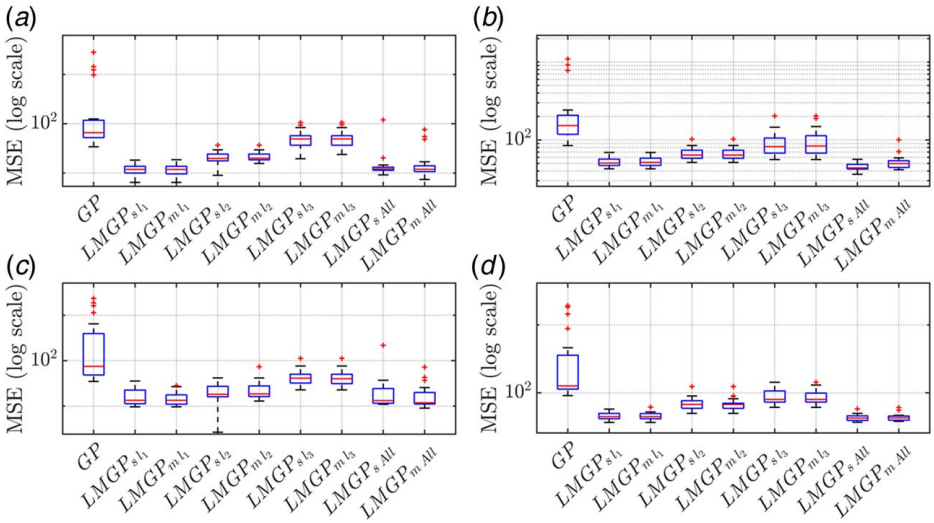

Fig. 13 High-fidelity emulation performance for wing weight example: Performance of the LMGP strategies follows the same trend as data source accuracy for all cases, with LMGP using only $y_{l_1}(\mathbf{x})$ arguably outperforming LMGP using all data sources. (*a*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 0$: LMGP using all data sources provides consistent estimates with some outliers. (*b*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 25$: $\text{LMGP}_{s\ All}$ performs noticeably better than other LMGP strategies for this case. (c) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 0$: LMGP using only $y_{l_1}(\mathbf{x})$ arguably outperforms LMGP using all data sources by a very slim margin. (*d*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 25$: Both LMGP strategies that use all data sources outperform those that only use $y_{l_1}(\mathbf{x})$ and $y_h(\mathbf{x})$ by a slim margin.

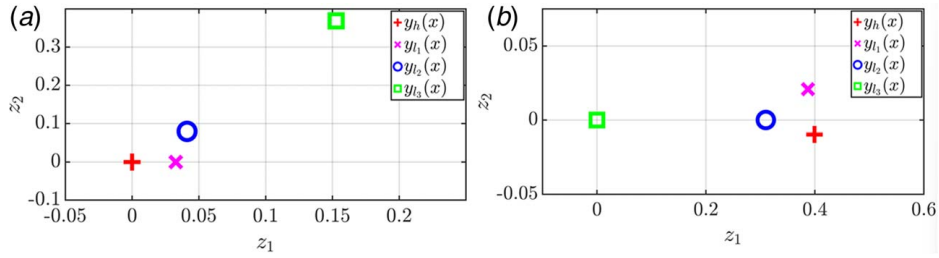

Fig. 14 Effect of constraints on the latent space: (*a*) Default constraints: The latent space for one sample repetition of LMGP fit all available data for the wing-weight function with $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 25$. $y_h(\boldsymbol{x})$, $y_{l_1}(\boldsymbol{x})$, and $y_{l_2}(\boldsymbol{x})$ are positioned at, respectively, the origin, positive z_1-axis, and first or second quadrant. While the learned latent spaces are different across the 30 repetitions, the relative latent distances are consistent both for different repetitions and for different amounts of data/noise. We only show the latent space of a randomly selected repetition. (*b*) Alternate constraints: The training procedure and data are exactly the same as before except that the three constrains are now applied to $y_{l_3}(\boldsymbol{x})$, $y_{l_2}(\boldsymbol{x})$, and $y_{l_1}(\boldsymbol{x})$. Note that the relative distances between data sources are the same between the two plots.

as follows: In the noiseless cases, LMGP is able to quite accurately learn the behavior of $y_h(\boldsymbol{x})$ using just $y_{l_1}(\boldsymbol{x})$ and using all four data sets provides no additional advantage in learning $y_h(\boldsymbol{x})$ while (1) requiring the estimation of additional hyperparameters (in the $\boldsymbol{A}$ matrix) and (2) compromising the estimates of $\boldsymbol{\Omega_x}$ to handle the discrepancies between the four sources. By contrast, in the cases with noise, one source is insufficient for LMGP to reach the threshold in emulation accuracy (which equals the noise variance) for $y_h(\boldsymbol{x})$. Including additional data sources in these cases helps LMGP to differentiate noise from model form error.

For the remainder of this example, we investigate the most challenging version which has the fewest available data and highest level of noise. The latent space for this problem for LMGP$_{s\ All}$, shown in Fig. 14(*a*), is once again a powerful diagnostic tool. While LMGP only has access to 15 noisy samples from the ten-dimensional function $y_h(\boldsymbol{x})$, the relative distances between latent positions match the relative accuracies of the data sources with respect to $y_h(\boldsymbol{x})$. The distance between $y_h(\boldsymbol{x})$ and $y_{l_3}(\boldsymbol{x})$ is ≈ 0.4 yielding an approximate correlation of $\exp\{-(0.4^2)\} \approx 0.85$, which means that LMGP still uses information from $y_{l_3}(\boldsymbol{x})$ in predicting the response for $y_h(\boldsymbol{x})$ despite the former's low accuracy with respect to the latter.

We impose a number of constraints in order to obtain a unique solution for the latent positions since our objective function in Eq. (8) is invariant under translation and rotation. For a two-dimensional latent space, we fix the first position to the origin, the second position to the positive z_1–axis, and the third position to the $z_2 > 0$ half-plane. As we mentioned before in Sec. 3.2, we also assign the data sources to positions sequentially (i.e., $[y_h(\boldsymbol{x}),\ y_{l_1}(\boldsymbol{x}),\ y_{l_2}(\boldsymbol{x}),\ y_{l_3}(\boldsymbol{x}),\ \cdots] \rightarrow [1,\ 2,\ 3,\ 4,\ \cdots]$) with $y_h(\boldsymbol{x})$ at the origin for easier visualization of the relative correlations $y_{l_i}(\boldsymbol{x})$. While assigning the data sources to latent positions affects the learned latent positions, the relative distances between them remain the same as shown in Fig. 14(*b*). Since we typically know the data source with the highest fidelity, the learned latent space of LMGP provides an extremely easy way to assess the fidelity of different sources with respect to it.

Prediction performance on the low-fidelity sources for LMGP$_{s\ All}$, shown in Fig. 15, follows the same trend as data source accuracy; i.e., it is best for $y_{l_1}(\boldsymbol{x})$ and worst for $y_{l_3}(\boldsymbol{x})$. When fitting LMGP to multiple data sources, we expect prediction accuracy to be high on sources that are well correlated with others, i.e., whose latent positions are close together or form a cluster. Leveraging information from a well-correlated source improves prediction performance more than the alternative, so each source in the cluster gains a boost in prediction performance from the information of the other sources in that cluster. In this case, $y_h(\boldsymbol{x})$, $y_{l_1}(\boldsymbol{x})$, and $y_{l_2}(\boldsymbol{x})$ form a cluster and as such we see that MSEs for $y_{l_1}(\boldsymbol{x})$ and $y_{l_2}(\boldsymbol{x})$ are much lower than those for $y_{l_3}(\boldsymbol{x})$.

In our next example, we consider data drawn from an eight-dimensional model of water flow through a borehole [46]:

$$y_h(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{2LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right)} \tag{20.1}$$

$$y_{l_1}(\boldsymbol{x}) = \frac{2\pi T_u (H_u - 0.8H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{1LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right)} \tag{20.2}$$

$$y_{l_2}(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{8LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + 0.75\frac{T_u}{T_l}\right)} \tag{20.3}$$

$$y_{l_3}(\boldsymbol{x}) = \frac{2\pi T_u (1.1H_u - H_l)}{\ln\left(\frac{4r}{r_w}\right)\left(1 + \frac{2LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right)} \tag{20.4}$$

$$\boldsymbol{x}^T = [T_u,\ H_u,\ H_l,\ r,\ r_w,\ T_l,\ L,\ K_w]$$

$$\min(\boldsymbol{x}) = [100,\ 990,\ 700,\ 100,\ 0.05,\ 10,\ 1000,\ 6000]$$

$$\max(\boldsymbol{x}) = [1000,\ 1110,\ 820,\ 10000,\ 0.15,\ 500,\ 2000,\ 12000]$$

The above equations indicate that all low-fidelity functions have nonlinear model form discrepancy. To roughly quantify these

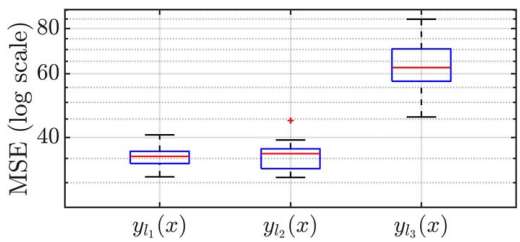

Fig. 15 Low-fidelity prediction performance: prediction accuracy is much higher for $y_{l_1}(\boldsymbol{x})$ and $y_{l_2}(\boldsymbol{x})$ than for $y_{l_3}(\boldsymbol{x})$

Table 3 Relative accuracy of functions for borehole problem

	$y_{l_1}(\boldsymbol{x})$	$y_{l_2}(\boldsymbol{x})$	$y_{l_3}(\boldsymbol{x})$
RRMSE	3.6671	1.3688	0.36232

Note: The functions are listed in increasing order with respect to accuracy, with $y_{l_3}(\boldsymbol{x})$ being the most accurate by a significant margin.

discrepancies, we follow the same procedure as in the previous example and calculate RRMSEs (Table 3). As it can be seen, the accuracy of the models increases with i (unlike the previous example—LMGP is robust with respect to this choice).

We consider various amounts of available low-fidelity data, with and without noise. We also use a few combinations for training LMGP based on the selected data sets or how data sources are encoded. The results are summarized in Fig. 16 where, once again, LMGP convincingly outperforms GP in high-fidelity emulation, especially with noisy data (Figs. 16(*b*) and 16(*d*)). The overall trends in performance between strategies for LMGP are consistent across the various cases, with LMGP fit to only one low-fidelity source performing worse than LMGP fit to all data sources and with $\text{LMGP}_{s\ All}$ specifically performing the best. $\text{LMGP}_{m\ All}$ yields inconsistent results with $n_l = 50$ or $n_l = 100$, especially in the latter case where the box plots have stretched to include the outliers. This behavior is due to overfitting and the fact that there are many latent positions that must be placed in the latent space via a simple matrix-based map (256 positions and 32 elements in the $\boldsymbol{A}$ matrix). Note that even with these inconsistencies, $\text{LMGP}_{m\ All}$ frequently outperforms GP, $\text{LMGP}_{s\ l_1}$, $\text{LMGP}_{m\ l_1}$, $\text{LMGP}_{s\ l_2}$, and $\text{LMGP}_{m\ l_2}$, which indicates that using more than two data sets in fusion is indeed beneficial.

The learned latent space for $\text{LMGP}_{s\ All}$ which is the most challenging version of this problem (noisy samples, fewest available data) is shown in Fig. 17(*a*) which clearly indicates that relative distances among the positions match with the relative accuracy between the low- and high-fidelity sources: The position for $y_{l_3}(\boldsymbol{x})$ is very close to that for $y_h(\boldsymbol{x})$, so LMGP weighs data from $y_{l_3}(\boldsymbol{x})$ heavily when emulating $y_h(\boldsymbol{x})$ and vice versa. The position for $y_{l_2}(\boldsymbol{x})$ is also close to both $y_h(\boldsymbol{x})$ and $y_{l_3}(\boldsymbol{x})$, but it is relatively more distant from $y_h(\boldsymbol{x})$ compared to $y_{l_3}(\boldsymbol{x})$.

Like in our first example, prediction performance on the low-fidelity sources for $\text{LMGP}_{s\ All}$, shown in Fig. 17(*b*), follows a similar trend to data source accuracy; i.e., it is best for $y_{l_2}(\boldsymbol{x})$ and $y_{l_3}(\boldsymbol{x})$ and worst for $y_{l_1}(\boldsymbol{x})$, which is the least accurate source. As we mentioned before, we expect prediction accuracy to be high on sources whose latent positions are close together or form a cluster. In this case, $y_h(\boldsymbol{x})$, $y_{l_2}(\boldsymbol{x})$, and $y_{l_3}(\boldsymbol{x})$ form a cluster, and as such, we see that MSEs for $y_{l_2}(\boldsymbol{x})$ and $y_{l_3}(\boldsymbol{x})$ are much lower than those for $y_{l_1}(\boldsymbol{x})$.

4.2 Calibration Results. We compare our calibration approach to that of KOH by considering four test cases with varying degrees of complexity. Note that, while LMGP can simultaneously assimilate and calibrate any number of sources, KOH's approach only works with two data sets at a time and relies on repeating the process as many times as there are low-fidelity sources.

For our first calibration problem, we consider data drawn from simple one-dimensional analytical functions

$$y_h(x) = \frac{1}{0.1x^3 + x^2 + x + 10}, \quad -2 \le x \le 3 \tag{21.1}$$

$$y_{l_1}(x) = \frac{1}{0.1x^3 + \theta x^2 + 1.5x + 10.5}, \quad -2 \le x \le 3 \quad \text{and} \quad -1 \le \theta \le 2 \tag{21.2}$$

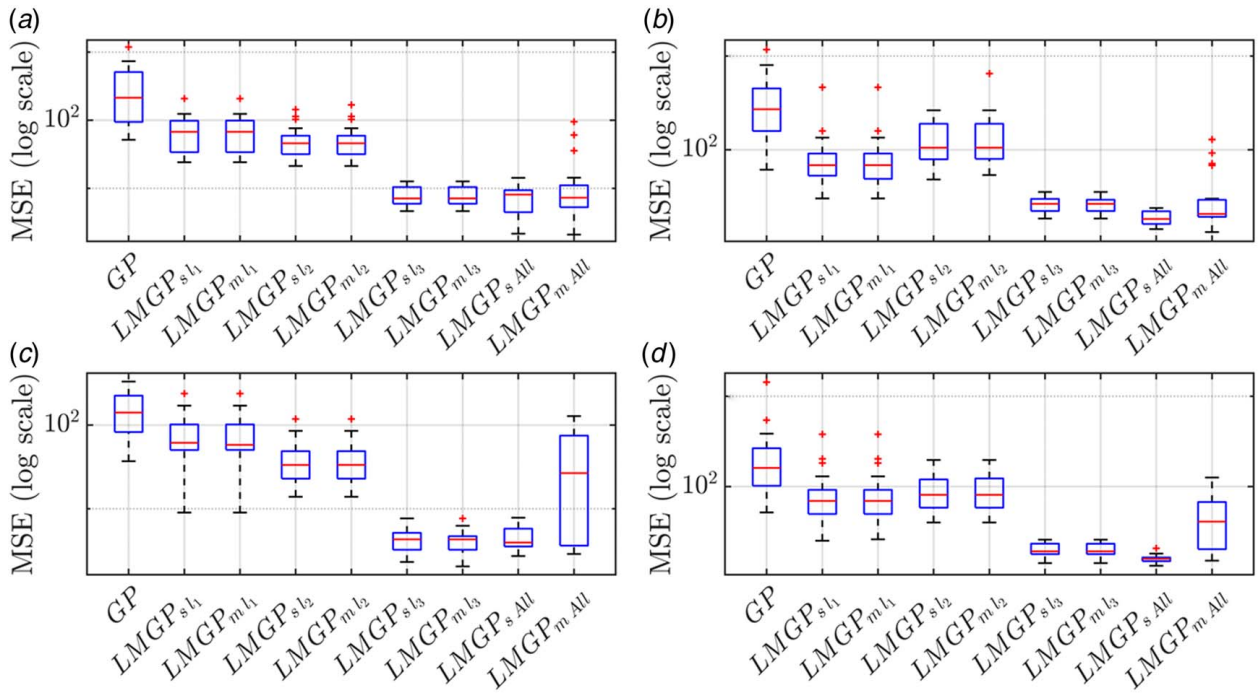

Fig. 16 High-fidelity emulation performance for the borehole problem: (*a*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 0$: LMGP strategies that use all data sources perform better than those using only one data source, with $\text{LMGP}_{s\ All}$ performing the best. (*b*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 6.25$: $\text{LMGP}_{s\ All}$ performs noticeably better than other LMGP strategies for this case. (*c*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 0$: $\text{LMGP}_{s\ All}$ again performs noticeably better than other LMGP strategies for this case. $\text{LMGP}_{m\ All}$ displays inconsistency in its estimates. (*d*) $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 6.25$: $\text{LMGP}_{s\ All}$ again performs noticeably better than other LMGP strategies for this case. $\text{LMGP}_{m\ All}$ again displays inconsistency in its estimates.

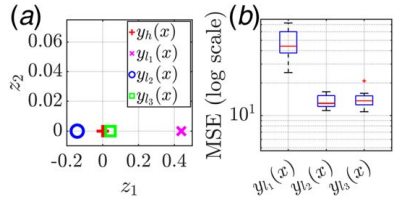

Fig. 17 Effects of correlations between data sources for borehole example: (*a*) Latent space: The latent space for one sample repetition of LMGP fit to all available data for the borehole function with $n_h = 15$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 6.25$. While the individual latent spaces are different for each repetition, the relative latent distances are consistent both for different repetitions and for different amounts of data/noise. (*b*) Low-fidelity MSEs: Low-fidelity prediction accuracy is better for $y_{l_2}(x)$ and $y_{l_3}(x)$ than for $y_{l_1}(x)$.

$$y_{l_2}(x) = \frac{1}{\theta x^2 + x + 10}, \quad -2 \le x \le 3 \quad \text{and} \quad -1 \le \theta \le 2 \qquad (21.3)$$

where we consider θ^*=1. Note that both low-fidelity sources have model form error, with $y_{l_2}(x)$ being more accurate than $y_{l_1}(x)$ over the input range when $\theta = \theta^*$ despite omitting the x^3 term (Table 4).

We show high-fidelity emulation performance for this problem in Fig. 18 where, similar to Sec. 4.1, LMGPs are trained under various settings in terms of which data sources are selected and how they are encoded. As it can be observed, LMGP performs on par with or better than KOH's approach in high-fidelity emulation accuracy for all cases, and LMGP$_{s\ All}$ offers the most consistent performance for most cases. LMGP also performs particularly well in the cases with noise (Figs. 18(*b*) and 18(*d*)). Despite the inaccuracy of $y_{l_2}(x)$, LMGP fit to all data sources offers the most accurate emulation in all cases.

Table 4 Relative accuracy of functions for simple calibration problem

	$y_{l_1}(x)$	$y_{l_2}(x)$
RRMSE	0.22241	0.1285

Note: We find the RRMSE in calibration problems using the same method as before but with the calibration parameters fixed to their true values at all input points. Both low-fidelity functions are relatively accurate, with $y_{l_2}(x)$ more accurate than $y_{l_1}(x)$.

We next show calibration performance in Fig. 19 where LMGP$_{s\ All}$ consistently outperforms KOH in both accuracy and consistency, especially in the noiseless cases (Figs. 19(*a*) and 19(*c*)). Notably, KOH's approach fit with $y_{l_2}(x)$ yields biased estimates. With noise and little data (Fig. 19(*b*)), neither LMGP nor KOH's approach are able to obtain a very consistent estimate for the calibration parameter across the repetitions. When more low-fidelity data are provided (Fig. 19(*d*)), LMGP is able to leverage the additional low-fidelity data to find a consistent estimate for θ while KOH's approach does not improve in consistency.

We show the latent space from fitting LMGP to the most challenging version of this problem, i.e., $n_h = 3$, $n_{l_1} = n_{l_2} = 15$, $\sigma^2 = 2 \times 10^{-5}$. As demonstrated in Fig. 20(*a*), LMGP is able to accurately infer the correlations with only three noisy high-fidelity samples as the relative latent distances match the relative accuracies of the data sources. Thus, we expect the low-fidelity performance to be better for $y_{l_2}(x)$ than for $y_{l_1}(x)$ as the position for $y_{l_2}(x)$ is relatively closer to $y_h(x)$, which means that LMGP leverages more information from $y_h(x)$ in predicting $y_{l_2}(x)$ than in predicting $y_{l_1}(x)$. We assess the veracity of our expectation by examining low-fidelity prediction performance in Fig. 20(*b*), which indicates that prediction performance is indeed better for $y_{l_2}(x)$ than for $y_{l_1}(x)$.

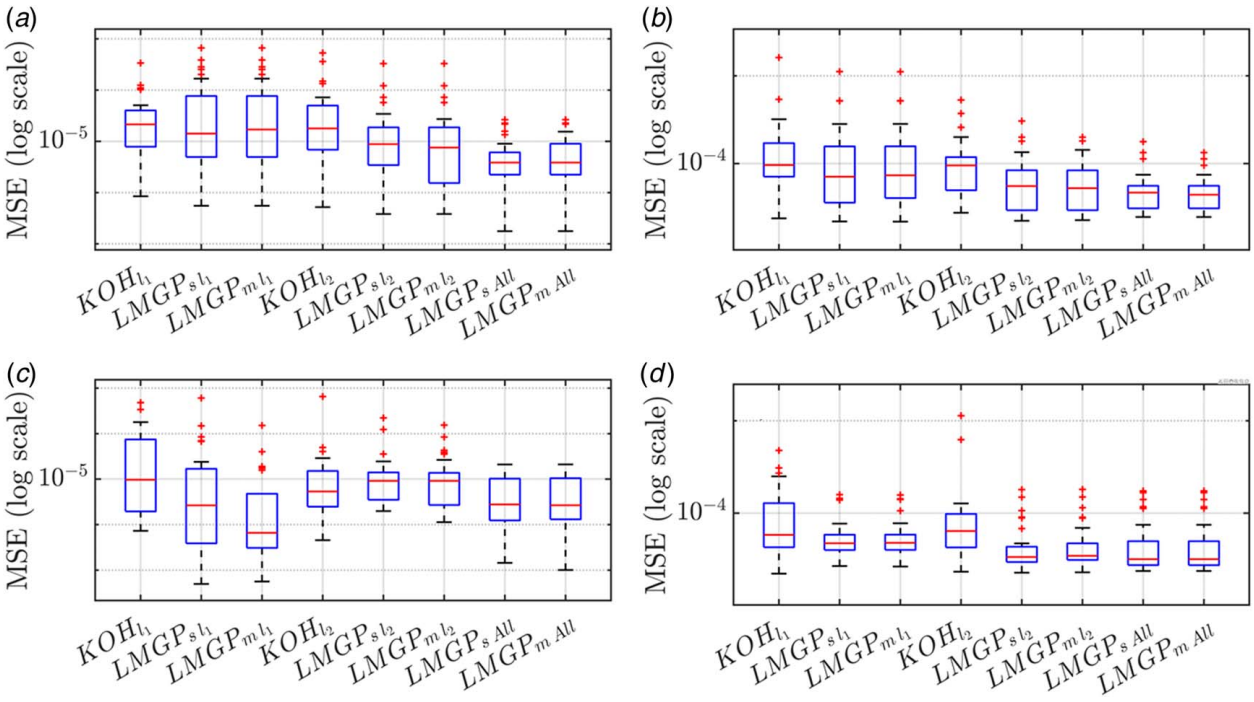

Fig. 18 High-fidelity emulation performance: (*a*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 15$, $\sigma^2 = 0$: LMGP strategies generally perform better than KOH's approach, with LMGP$_{s\ All}$ performing the best. Estimates for all strategies except LMGP$_{s\ All}$ are fairly inconsistent. (*b*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 15$, $\sigma^2 = 2 \cdot 10^{-5}$: LMGP$_{s\ All}$ performs noticeably better than other LMGP strategies for this case (and better than KOH's approach). (*c*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 0$: With the addition of more low-fidelity data, all approaches perform better. LMGP$_{s\ All}$ performs best by a very slim margin, and is more consistent in its performance than comparable strategies. (*d*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 2 \cdot 10^{-5}$: With noise, LMGP$_{s\ l_2}$ performs nearly on par with LMGP$_{s\ All}$ and produces more consistent performance.

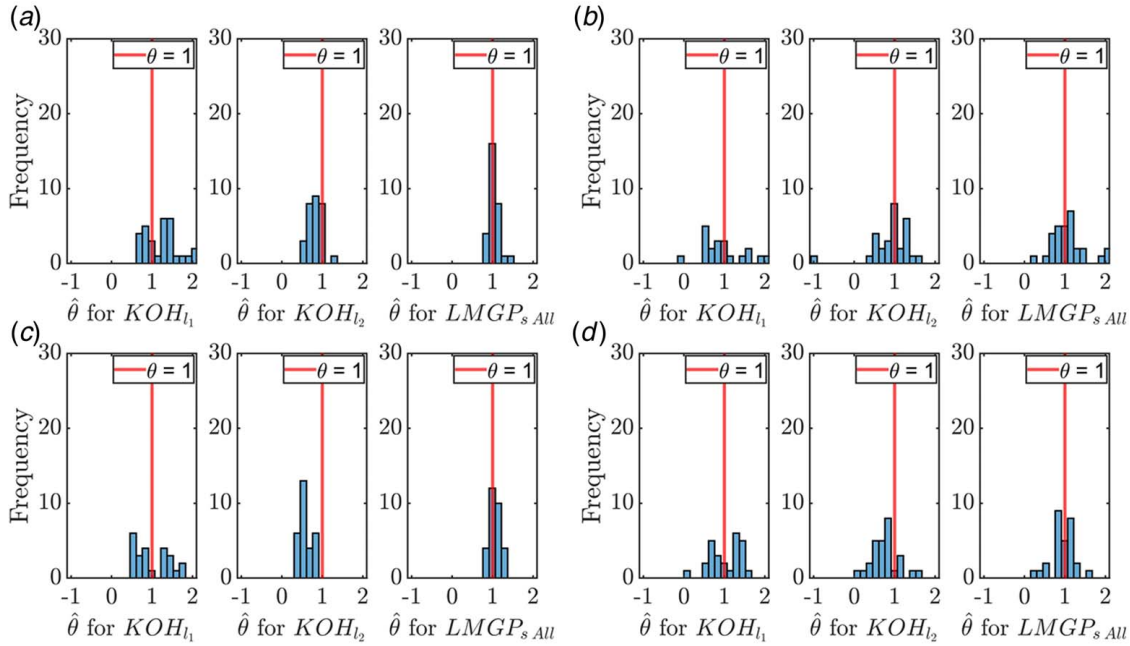

Fig. 19 Calibration performance: (*a*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 15$, $\sigma^2 = 0$: LMGP offers consistent and unbiased estimates. KOH's approach suffers from bias and inconsistency. (*b*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 15$, $\sigma^2 = 2 \cdot 10^{-5}$: All approaches yield inconsistent estimates. (*c*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 0$: Both KOH's approach and LMGP yield consistent estimates, but KOH's approach still suffers from bias. (*d*) $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 50$, $\sigma^2 = 2 \cdot 10^{-5}$: LMGP achieves higher consistency that KOH's approach with the addition of more low-fidelity data. LMGP's estimate is unbiased, while KOH's approach still yields biased estimates.

Next, we reconsider the example in Eq. (18) where $\theta^* = \pi$ and $\theta^* = 10\pi$ are the two valid choices for the true calibration parameter as discussed in Sec. 3.3. We fit LMGP with two approaches to categorical variable selection and consider various amounts of available low-fidelity data all with noise (the noiseless case is considered in Sec. 3.3).

The high-fidelity emulation performance is summarized in Fig. 21, which indicates that LMGP outperforms KOH's approach by a similar margin for each case. Notably, LMGP's performance is robust to the choice of categorical variable assignment for this problem as we see a similar variation in performance over repetitions between $\text{LMGP}_{s\ All}$ and $\text{LMGP}_{m\ All}$. We explain this by noting that since there are only two data sources, $\text{LMGP}_{m\ All}$ finds a total of $2^2 = 4$ latent positions with $(2+2) \times 2 = 8$ elements in $\boldsymbol{A}$ which indicates that overfitting should not be a concern.

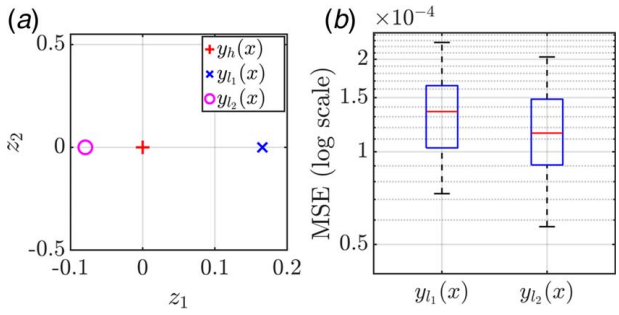

Fig. 20 Effects of correlations between data sources: (*a*) Latent space: The latent space for one sample repetition of LMGP fit to all available data with $n_h = 3$, $n_{l_1} = n_{l_2} = n_{l_3} = 15$, $\sigma^2 = 2 \times 10^{-5}$. While the individual latent spaces are different for each repetition, the relative latent distances are consistent both for different repetitions and for different amounts of data/noise. (*b*) Low-fidelity MSEs: Low-fidelity prediction accuracy is better for $y_{l_2}(x)$ than for $y_{l_1}(x)$.

The estimates of the calibration parameters are provided in Fig. 22 and indicate that the estimation consistency in both approaches increases as n_l is increased from 30 to 200. This increase is more prominent for LMGP. However, while LMGP converges on $\theta = 10\pi$, KOH's approach's estimates are approximately evenly split between π and 10π. This behavior is because the $L2$ distance of $\sin(10\pi x)$ and $\sin(\pi x)$ from $y_h(x)$ is the same, and hence, KOH's approach cannot favor one over the other [21,47,48]. As explained in Sec. 3.3, in this case, LMGP converges at $\theta = 10\pi$ as this choice provides not only a simpler discrepancy but also enables learning the high-frequency nature of $y_h(x)$.

Finally, we show histograms of latent distances learned by LMGP in Fig. 23. The trends are quite similar to those seen in Sec. 3.3, with the latent distances being close to 0 for low amounts of low-fidelity data and converging on 0.5 as the amount of data is increased. When high-fidelity data are insufficient to learn the high-frequency behavior of $y_h(x)$, LMGP treats the high-frequency behavior as noise and finds $y_h(x) \approx \sin(\pi x)$. When low-fidelity data are also insufficient, LMGP cannot learn the behavior of $y_l(x)$ at high frequencies (i.e., for large θ). Thus, LMGP finds $\theta = \pi$, which implies $y_l(x) = \sin(\pi x)$, i.e., no model form error and a corresponding latent distance near zero. With sufficient low-fidelity data, however, LMGP learns the behavior of $y_l(x)$ for large θ and finds that $\theta = 10\pi$ yields a less complex discrepancy between $y_h(x)$ and $y_l(x)$.

We now revisit the borehole problem from Sec. 4.1, this time adapted as a calibration problem to explore the effects of both nonlinear model form error and high-dimensional inputs. We begin with data drawn from the following functions:

$$y_h(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{2 \cdot 1500 \cdot T_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{250}\right)} \tag{22.1}$$

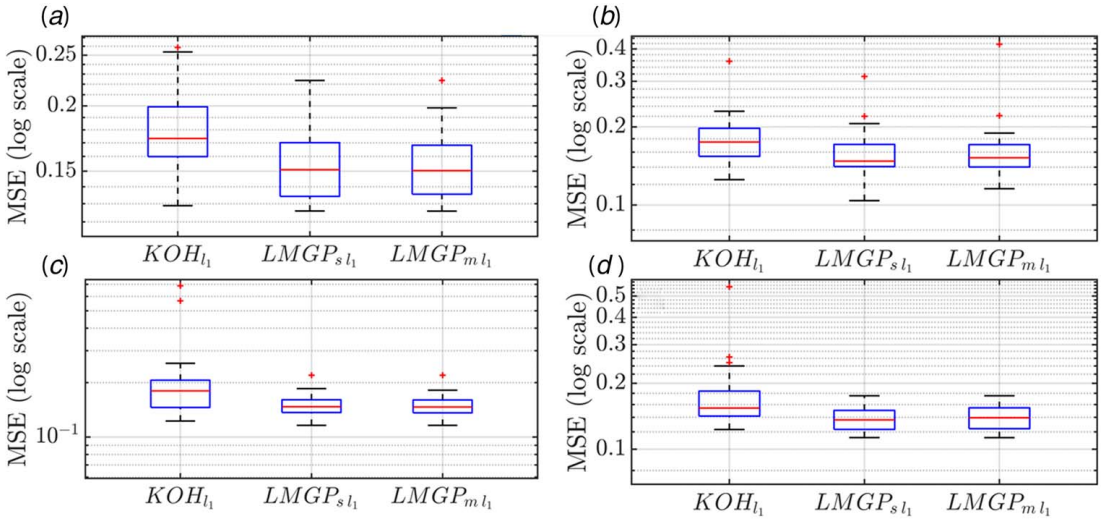

Fig. 21 High-fidelity emulation performance for sin wave example: (*a*) $n_h = 30$, $n_l = 30$, $\sigma^2 = .09$, (*b*) $n_h = 30$, $n_l = 60$, $\sigma^2 = .09$, (c) $n_h = 30$, $n_l = 100$, $\sigma^2 = .09$, and (*d*) $n_h = 30$, $n_l = 200$, $\sigma^2 = 0.09$. LMGP outperforms KOH's approach by a similar margin in all cases.

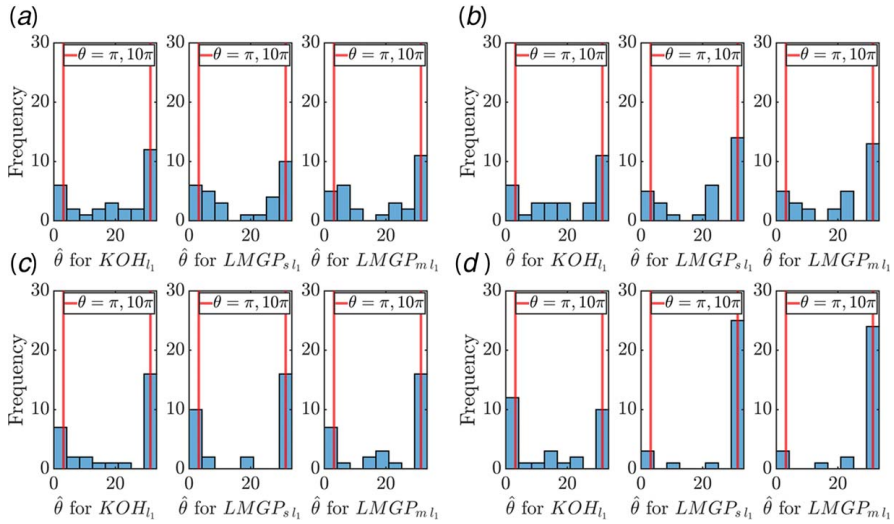

Fig. 22 Calibration performance for sin wave problem: (*a*) $n_h = 30$, $n_l = 30$, $\sigma^2 = .09$, (*b*) $n_h = 30$, $n_l = 60$, $\sigma^2 = .09$, (c) $n_h = 30$, $n_l = 100$, $\sigma^2 = .09$, and (*d*) $n_h = 30$, $n_l = 200$, $\sigma^2 = .09$

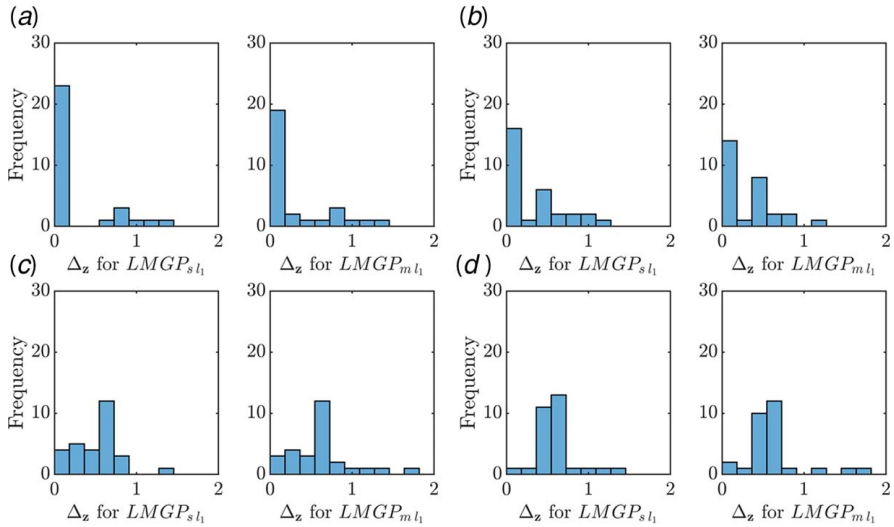

Fig. 23 Histogram of latent distances for sin wave problem: (*a*) $n_h = 30$, $n_l = 30$, $\sigma^2 = .09$, (*b*) $n_h = 30$, $n_l = 60$, $\sigma^2 = .09$, (c) $n_h = 30$, $n_l = 100$, $\sigma^2 = .09$, (*d*) $n_h = 30$, $n_l = 200$, $\sigma^2 = .09$

Table 5 Relative accuracy of functions for borehole calibration problem

	$y_{l_1}(\boldsymbol{x})$	$y_{l_2}(\boldsymbol{x})$
RRMSE	0.049219	0.19838

Note: Both low-fidelity functions are relatively accurate, with $y_{l_2}(x)$ less accurate than $y_{l_1}(x)$.

$$y_{l_1}(\boldsymbol{x}) = \frac{2\pi \times 500 \times (0.993 \times H_u - H_l)}{0.95 \times \ln\left(\frac{r}{r_w}\right)\left(1 + \frac{2\theta_2 \times 500}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{500}{\theta_1}\right)} \tag{22.2}$$

$$y_{l_2}(\boldsymbol{x}) = \frac{2\pi \times 500 \times (H_u - 1.045 \times H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{2\theta_2 \times 500}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{500}{\theta_1}\right)} \tag{22.3}$$

$$\boldsymbol{x}^T = [T_u,\ H_u,\ H_l,\ r,\ r_w,\ K_w],\ \boldsymbol{\theta}^T = [\theta_1,\ \theta_2]$$

$$\min(\boldsymbol{x}) = [100,\ 990,\ 700,\ 100,\ 0.05,\ 6000]$$

$$\max(\boldsymbol{x}) = [1000,\ 1110,\ 820,\ 10000,\ 0.15,\ 12000]$$

$$\min(\boldsymbol{\theta}^T) = [10,\ 1000],\quad \max(\boldsymbol{\theta}^T) = [500,\ 2000]$$

where we consider $\boldsymbol{\theta}^{T^*}$=[250,1500]. Note that both low-fidelity sources have model form error with $y_{l_1}(\boldsymbol{x})$ being more accurate than $y_{l_2}(\boldsymbol{x})$ over the input range when the true calibration parameters are used (Table 5) and that the input T_u has been omitted and replaced by a constant in both low-fidelity functions.

We hold $n_h = 25$ and $n_l = 100$ constant and examine two cases, one without noise and one with noise applied to samples ($\sigma^2 = 100$ with Range($y_h(\boldsymbol{x})$) ≈ 974 over the input range) and again fit LMGP with various strategies. In both cases, LMGP convincingly outperforms KOH's approach in high-fidelity emulation, see Fig. 24. Notably, LMGP outperforms KOH's approach given equivalent access to data, e.g., LMGP$_{s\,l_1}$ versus KOH$_{l_1}$. LMGP's performance is also robust to modeling choice, which we explain by noting that with three data sources the $\boldsymbol{t}_m$ strategy for categorical variable selection yields $3^3 = 27$ latent positions and $2 \times (3 \times 3) = 18$ elements of $\boldsymbol{A}$, i.e., the number of latent positions is on the same order of magnitude as the number of hyperparameters in $\boldsymbol{A}$ and the size of the dataset is large relative to the number of hyperparameters.

As shown in Fig. 25(*a*) for the noiseless case, the latent positions found by LMGP$_{s\,All}$ show no model form error for $y_{l_1}(\boldsymbol{x})$ and little model form error for $y_{l_2}(\boldsymbol{x})$, i.e., LMGP mistakes model form error in $y_{l_1}(\boldsymbol{x})$ for noise since the error is so low. While these latent positions are not fully accurate as $y_{l_1}(\boldsymbol{x})$ does still have model form error, the relative distances to the data sources do correctly indicate which is more accurate. With noise, shown in Fig. 25(*b*), the relative distances to $y_h(\boldsymbol{x})$ are nearly the same for both low-fidelity sources, although $y_{l_1}(\boldsymbol{x})$ is slightly closer to $y_h(\boldsymbol{x})$ than to $y_{l_2}(\boldsymbol{x})$, which indicates that LMGP has more difficulty determining the magnitudes of the errors in the low-fidelity data sources in this case. The magnitudes of the latent distances are quite small in both cases, which reflect the fact that both low-fidelity data sources are relatively accurate when calibrated appropriately.

Calibration performance, shown in Fig. 26, reveals inconsistent performance in estimating θ_1 but consistent estimates for θ_2 for both LMGP and KOH's approach in all three cases. We explain this by noting that the main sensitivity indices (calculated using 10, 000 inputs sampled via Sobol sequence) for θ_1 and θ_2 are on the order of 10^{-4} and 10^{-1} respectively for the low-fidelity functions; i.e., variation in θ_1 has very little effect on their outputs. Therefore, we expect θ_1 to be very difficult to estimate. While

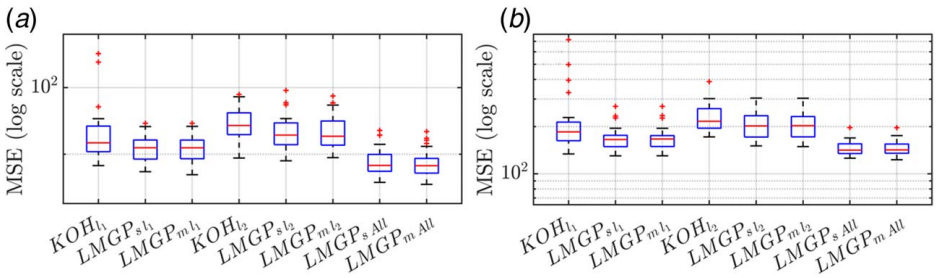

Fig. 24 High-fidelity emulation performance: (*a*) $n_h = 25$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 0$: LMGP$_{m\,All}$ arguably performs better than LMGP$_{s\,All}$. (*b*) $n_h = 25$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 100$: Results with noise are quite similar to those without.

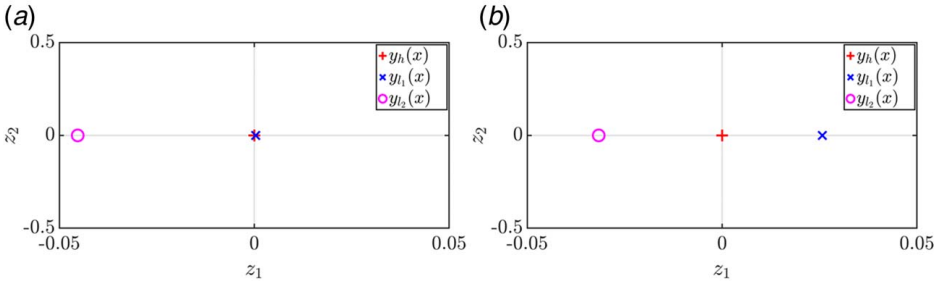

Fig. 25 Latent positions: (*a*) $n_h = 25$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 0$: LMGP finds no model form error for $y_{l_2}(x)$ and instead mistakes it for noise. (*b*) $n_h = 25$, $n_{l_1} = n_{l_2} = n_{l_3} = 100$, $\sigma^2 = 100$: LMGP correctly finds little error for both sources, but is unable to accurately determine the relative magnitudes of those errors.

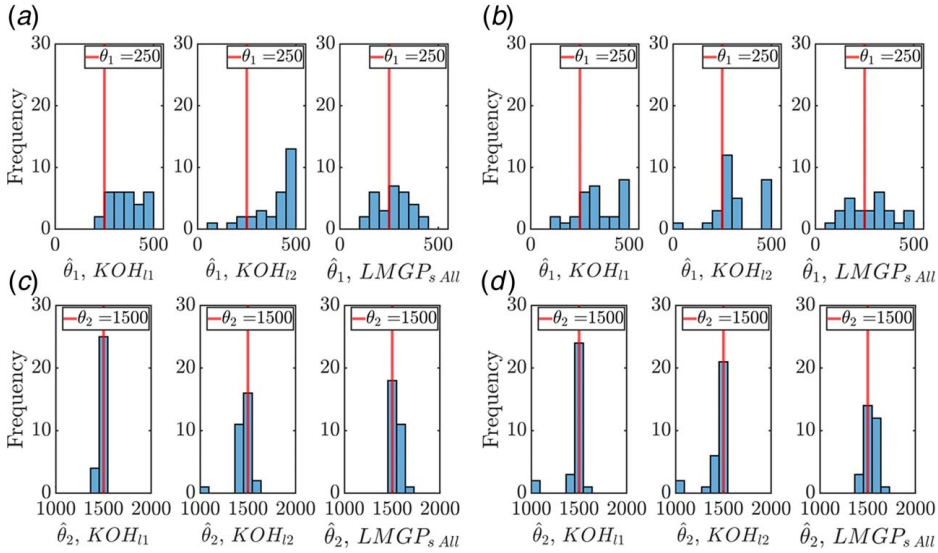

Fig. 26 Calibration performance: (*a*) $\hat{\theta}_1$ for $n_h = 25$, $n_{l_1} = n_{l_2} = 100$, $\sigma^2 = 0$: KOH's approach produces biased estimates, while LMGP's estimates are centered on the correct parameter with high variance. (*b*) $\hat{\theta}_1$ for $n_h = 25$, $n_{l_1} = n_{l_2} = 100$, $\sigma^2 = 100$: KOH's approach again produces biased estimates, with the caveat that KOH_{l_2} finds the correct parameter nearly half the time. (*c*) $\hat{\theta}_2$ for $n_h = 25$, $n_{l_1} = n_{l_2} = 100$, $\sigma^2 = 0$: All methods find the correct parameter consistently. KOH_{l_1} finds the most accurate and consistent estimates, while KOH_{l_2} has some outliers. (*d*) $\hat{\theta}_2$ for $n_h = 25$, $n_{l_1} = n_{l_2} = 100$, $\sigma^2 = 100$: Both of KOH's approaches have outliers, but estimate the correct parameter more consistently than LMGP.

LMGP's estimates for θ_1 suffer from high variance, the distributions are centered on the true parameter for both cases. By contrast, KOH's approach produces biased estimates in all cases, although KOH_{l_2} guesses nearly the correct parameter almost half the time in the case with noise (Fig. 26(*b*)). Both methods estimate θ_2 quite accurately and consistently. KOH's approach has lower variance in its estimates but more outliers when using $y_{l_2}(\boldsymbol{x})$ compared to LMGP's estimates using all data sources.

Finally, we revisit the wing-weight problem from Sec. 4.1, now adapted as a calibration problem with four calibration parameters. We begin with data drawn from the following functions:

$$y_h(\boldsymbol{x}) = 0.036 S_\omega^{0.758} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} \theta_1^{0.006} \theta_2^{0.04} \left(\frac{100\theta_3}{\cos(\Lambda)}\right)^{-0.3} (\theta_4 W_{dg})^{0.49} + S_\omega W_p \tag{23.1}$$

$$y_{l_1}(\boldsymbol{x}) = 0.036 S_\omega^{0.758} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} \theta_1^{0.006} \theta_2^{0.04} \left(\frac{100\theta_3}{\cos(\Lambda)}\right)^{-0.3} (\theta_4 W_{dg})^{0.49} + 1 \times W_p \tag{23.2}$$

$$y_{l_2}(\boldsymbol{x}) = 0.036 S_\omega^{0.8} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} \theta_1^{0.006} \theta_2^{0.04} \left(\frac{100\theta_3}{\cos(\Lambda)}\right)^{-0.3} (\theta_4 W_{dg})^{0.49} + 1 \times W_p \tag{23.3}$$

$$y_{l_3}(\boldsymbol{x}) = 0.036 S_\omega^{0.9} W_{f\omega}^{0.0035} \left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} \theta_1^{0.006} \theta_2^{0.04} \left(\frac{100\theta_3}{\cos(\Lambda)}\right)^{-0.3} (\theta_4 W_{dg})^{0.49} + 0 \times W_p \tag{23.4}$$

$$\boldsymbol{x}^T = [S_\omega,\ W_{f\omega},\ A,\ \Lambda,\ W_{dg},\ W_p],\ \boldsymbol{\theta}^T = [\theta_1,\ \theta_2, \theta_3, \theta_4]$$

$$\min(\boldsymbol{x}) = [150,\ 220,\ 6,\ -10,\ 1700,\ 0.025]$$

$$\max(\boldsymbol{x}) = [200,\ 300,\ 10,\ 10,\ 2500,\ 0.08]$$

$$\min(\boldsymbol{\theta}^T) = [16,\ 0.5,\ 0.08,\ 2.5],\quad \max(\boldsymbol{\theta}^T) = [45,\ 1,\ 0.18,\ 6]$$

where we consider $\boldsymbol{\theta}^{T^*}$=[40,0.8,0.17,3]. Note that all three low-fidelity sources have model form error that increases with the data

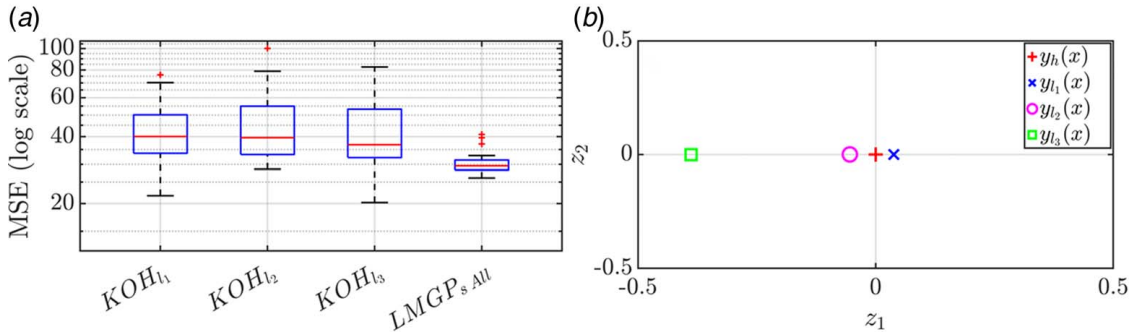

Fig. 27 Analysis for the Wing-Weight Problem: (*a*) high-fidelity emulation performance: LMGP displays much more consistency than KOH's method. (*b*) latent space: The latent space accurately reflects the relative accuracies of the data sources.

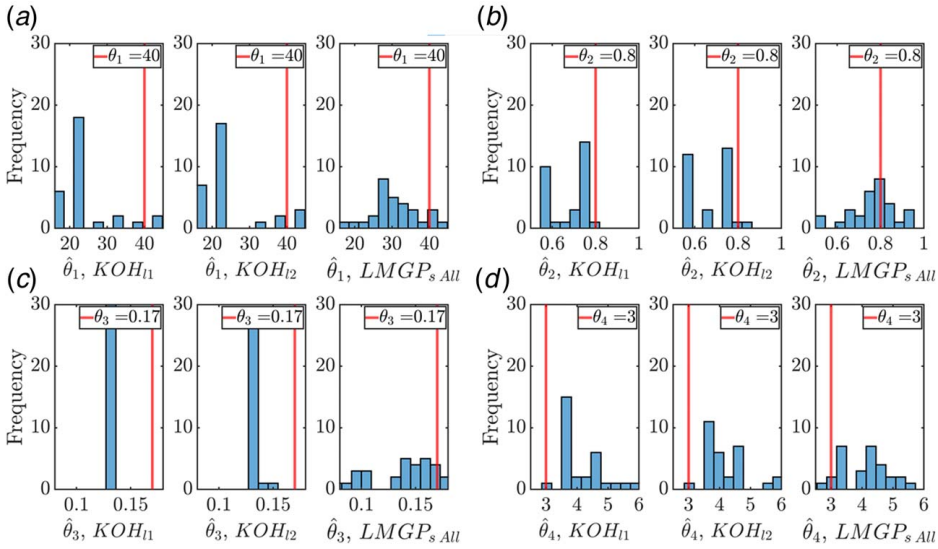

Fig. 28 Calibration Performance for the Wing-Weight Problem: We do not show the calibration results for KOH_{l_3} as they are very poor due to the low accuracy of $y_{l_3}(x)$

source index when the true calibration parameters are used (Table 2).

We examine one case with very small noisy data sets in which we set $n_h = 15$, $n_l = 50$, and $\sigma^2 = 16$. We fit only $LMGP_{s\,All}$ as it is generally the best-performing model. LMGP consistently outperforms KOH's method in high-fidelity emulation (Fig. 27(*a*)). Additionally, the latent space learned by LMGP shows model form error for all three low-fidelity sources, with the relative distances between the sources roughly matching their relative accuracies (Fig. 27(*b*)). Both KOH's method and LMGP perform poorly in calibration for all four parameters. We explain this by noting that this problem suffers from identifiability issues and that the calibration parameters have both small Sobol sensitivities and low interaction (i.e., even increasing the number of data points will not resolve the issue). Notably, while LMGP displays inconsistent calibration estimates for each parameter, KOH's method incorrectly shows consistent but biased estimates which are often quite far from the true calibration parameter (Fig. 28(*c*)). LMGP shows more uncertainty than KOH, which more accurately reflects the nature of the problem and its learned latent space can help the analyst in detecting identifiability issues.

5 Conclusion

In this paper, we present a novel latent-space-based approach for data fusion (i.e., multi-fidelity modeling and calibration) via latent-map Gaussian processes or LMGPs. Our approach offers unique advantages that can benefit engineering design in a number of ways such as improved accuracy and consistency compared to competing methods for data fusion. Additionally, LMGP learns a latent space where data sources are embedded with points whose distances can shed light on not only the relations among data sources but also potential model form discrepancies. These insights can guide diagnostics or determine which data sources cannot be trusted.

Implementation and use of our data fusion approach are quite straightforward as it primarily relies on modifying the correlation function of traditional GPs and assigning appropriate priors to the datasets. LMGP-based data fusion is also quite flexible in terms of the number of data sources. In particular, since we can assimilate multiple data sets simultaneously, we improve prediction performance and decrease non-identifiability issues that typically arise in calibration problems.

Since LMGPs are extensions of GPs, they are not directly applicable to extrapolation or big/high-dimensional data. However, extensions of GPs that address these limitations [27,38,41–44,49] can be incorporated into LMGPs. In our examples, we assumed all data sources are noisy and hence used a single parameter to estimate the noise. To consider different (unknown) noise levels, we need to have a parameter for each data source. We also note that the performance of LMGP in fusing small data can be greatly improved by endowing its parameters with priors and using Bayes' rule for inference. In this case, the latent space will have a probabilistic nature, the trained model will be more robust to overfitting, and prediction uncertainties will be more accurate. Lastly, we have studied small data scenarios and not explored the effects of large data sets on the consistency of hyperparameter estimation. A detailed convergence study is needed to determine how the hyperparameters and the learned manifold are affected as the data set sizes grow. These directions will be investigated in our future works.

Lastly, we note that the proposed method can be directly applied to multi-response data sets with no modifications. To apply LMGP, we would treat each response as if it was a data source and then apply our data fusion method directly. However, with this strategy, each "data source' would have the exact same set of input points, which will most likely cause numerical issues. While LMGP can be applied to multi-response data sets with some modifications (which may be presented in a future paper), the user should bear in mind that we do not necessarily *a priori* expect any level of correlation between the responses whereas with multi-fidelity problems we expect (but do not necessarily have) some correlation as all sources model the same system. Thus, we would recommend fitting LMGP to all responses and examining the latent space to see which responses are well-correlated. Then, fit individual emulators to uncorrelated responses while fitting an LMGP to whichever groups of responses that are correlated with each other.

Acknowledgment

This work was supported by the Early Career Faculty grant from NASA's Space Technology Research Grants Program (Award Number 80NSSC21K1809).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data sets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Nomenclature

$\boldsymbol{t}$ = matrix or vector encoding of the categorical combinations used in LMGP

$\boldsymbol{A}$ = matrix of hyperparameters of LMGP which determine the latent positions of the categorical combinations

$\boldsymbol{R}$ = correlation matrix for LMGP

n_h, n_{l_i} = respectively, the number of training data for the high-fidelity source and ith low-fidelity source. When all low-fidelity sources have the same number of training data, we simply use n_l

$\boldsymbol{y}_h$, y_h^i, $y_h(\boldsymbol{x})$, y_h = respectively, a vector containing training outputs, the ith training output, the underlying data source, and the output of the underlying data source

$\boldsymbol{X}_h$, $\boldsymbol{x}_h^i$, $\boldsymbol{x}_h$ = respectively, the matrix of training inputs, the ith training input, and the input to the data source. In the case that the input is one-dimensional, these become $\boldsymbol{x}_h$, x_h^i, and x_h, respectively

$\Delta_{z_{y_h}, z_{y_{l_i}}}$ = distance between, e.g., $y_h(\boldsymbol{x})$ and $y_{l_i}(\boldsymbol{x})$ in the latent space. In the case that there are only two points in the latent space, we shorten this to just Δ_z

$\boldsymbol{\theta}$, $\boldsymbol{\theta}^*$, $\hat{\boldsymbol{\theta}}$ = respectively, the calibration inputs, true calibration parameters, and estimated calibration parameters. In general, we use an asterisk to denote the true value of a parameter and a hat to denote an estimate

σ^2 = noise variance

$\boldsymbol{\Omega}_{\boldsymbol{x}}$, $\boldsymbol{\Omega}_{\boldsymbol{\theta}}$ = matrix of roughness parameters ω_i for the numerical and calibration inputs, respectively

Subscripts

h = high-fidelity source

l_i = ith low-fidelity source. We use this and the above subscript to denote data sources and their corresponding latent points, e.g., $y_h(\boldsymbol{x})$ or $y_{l_1}(\boldsymbol{x})$. We also use this subscript to refer to strategies of KOH's approach or LMGP which are fit to only $\boldsymbol{y}_h$ and $\boldsymbol{y}_{l_i}$

s, m = respectively, strategy 1 and strategy 2 for categorical variable assignment during preprocessing of data for LMGP. We combine this with the above subscripts to fully describe a fitting strategy, e.g., $\text{LMGP}_{s\,l_1}$ denotes LMGP fit to only $\boldsymbol{y}_h$ and $\boldsymbol{y}_{l_1}$ using strategy 1 for preprocessing the data

All = a strategy of LMGP fit to data from all available sources

References

[1] Chaudhuri, A., Lam, R., and Willcox, K., 2018, "Multifidelity Uncertainty Propagation via Adaptive Surrogates in Coupled Multidisciplinary Systems," AIAA J., **56**(1), pp. 235–249.

[2] Peherstorfer, B., Willcox, K., and Gunzburger, M., 2018, "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," SIAM Rev., **60**(3), pp. 550–591.

[3] Kennedy, M. C., and O'Hagan, A., 2001, "Bayesian Calibration of Computer Models," J. R. Stat. Soc. Series B Stat. Methodol., **63**(3), pp. 425–464.

[4] Tao, S., Apley, D. W., Chen, W., Garbo, A., Pate, D. J., and German, B. J., 2019, "Input Mapping for Model Calibration With Application to Wing Aerodynamics," AIAA J., **57**(7), pp. 2734–2745.

[5] Koziel, S., Cheng, Q. S., and Bandler, J. W., 2008, "Space Mapping," IEEE Microwave Mag., **9**(6), pp. 105–122.

[6] Bandler, J. W., Biernacki, R. M., Chen, S. H., Grobelny, P. A., and Hemmers, R. H., 1994, "Space Mapping Technique for Electromagnetic Optimization," IEEE Trans. Microwave Theory Tech., **42**(12), pp. 2536–2544.

[7] Amrit, A., Leifsson, L., and Koziel, S., 2020, "Fast Multi-Objective Aerodynamic Optimization Using Sequential Domain Patching and Multifidelity Models," J. Aircr., **57**(3), pp. 388–398.

[8] Leifsson, L., and Koziel, S., 2015, "Aerodynamic Shape Optimization by Variable-Fidelity Computational Fluid Dynamics Models: A Review of Recent Progress," J. Comput. Sci., **10**, pp. 45–54.

[9] Koziel, S., and Leifsson, L., 2013, "Multi-Level CFD-Based Airfoil Shape Optimization With Automated Low-Fidelity Model Selection," Procedia Comput. Sci., **18**(1), pp. 889–898.

[10] Forrester, A., Sobester, A., and Keane, A., 2008, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, Hoboken, NJ.

[11] Ng, L. W.-T., and Eldred, M., 2012, "Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation," Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, Honolulu, HI, Apr. 23–26.

[12] Padron, A. S., Alonso, J. J., and Eldred, M. S., 2016, "Multi-Fidelity Methods in Aerodynamic Robust Optimization," Proceedings of the 18th AIAA non-Deterministic Approaches Conference., San Diego, CA, Jan. 4–8.

[13] Zadeh, P. M., Toropov, V. V., and Wood, A. S., 2005, "Use of Moving Least Squares Method in Collaborative Optimization," Proceedings of the 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, May 30–June 3.

[14] Fernández-Godino, M. G., Park, C., Kim, N. H., and Haftka, R. T., 2016, "Review of Multi-Fidelity Models," arXiv preprint.

[15] Romanowicz, R., Beven, K. J., and Tawn, J., 1994, "Evaluation of Predictive Uncertainty in Nonlinear Hydrological Models Using a Bayesian Approach," Stat. Environ., **2**, pp. 297–317.

[16] Craig, P. S., Goldstein, M., Rougier, J. C, and Seheult, A. H., 2001, "Bayesian Forecasting Using Large Computer Models," J. Am. Stat. Assoc., **96**(454), pp. 717–729.

[17] Stainforth, D. A., Aina, T., Christensen, C., Collins, M., Faull, N., Frame, D. J., Kettleborough, J. A., et al., 2005, "Uncertainty in Predictions of the Climate Response to Rising Levels of Greenhouse Gases," Nature, **433**(7024), pp. 403–406.

[18] Zhang, W., Bostanabad, R., Liang, B., Su, X., Zeng, D., Bessa, M. A., Wang, Y., Chen, W., and Cao, J., 2019, "A Numerical Bayesian-Calibrated Characterization Method for Multiscale Prepreg Preforming Simulations With Tension-Shear Coupling," Compos. Sci. Technol., **170**(C), pp. 15–24.

[19] Gramacy, R. B., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., Rutter, E., Trantham, M., and Drake, R. P., 2015, "Calibrating a Large Computer Experiment Simulating Radiative Shock Hydrodynamics," Ann. Appl. Stat., **9**(3), pp. 1141–1168.

[20] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D., 2004, "Combining Field Data and Computer Simulations for Calibration and Prediction," SIAM J. Sci. Comput., **26**(2), pp. 448–466.

[21] Plumlee, M., 2017, "Bayesian Calibration of Inexact Computer Models," J. Am. Stat. Assoc., **112**(519), pp. 1274–1285.

[22] Apley, D. W., Liu, J., and Chen, W., 2006, "Understanding the Effects of Model Uncertainty in Robust Design With Computer Experiments," ASME J. Mech. Des., **128**(4), pp. 945–958.

[23] Bayarri, M. J., Berger, J. O, Paulo, R., Sacks, J., Cafeo, J. A, Cavendish, J., Lin, C.-H., and Tu, J., 2007, "A Framework for Validation of Computer Models," Technometrics, **49**(2), pp. 138–154.

[24] Arendt, P. D., Apley, D. W., and Chen, W., 2012, "Quantification of Model Uncertainty: Calibration, Model Discrepancy, and Identifiability," ASME J. Mech. Des., **134**(10), p. 100908.

[25] Arendt, P. D., Apley, D. W., Chen, W., Lamb, D., and Gorsich, D., 2012, "Improving Identifiability in Model Calibration Using Multiple Responses," ASME J. Mech. Des., **134**(10), p. 100909.

[26] Bostanabad, R., Kearney, T., Tao, S., Apley, D. W., and Chen, W., 2018, "Leveraging the Nugget Parameter for Efficient Gaussian Process Modeling," Int. J. Numer. Methods Eng., **114**(5), pp. 501–516.

[27] Rasmussen, C. E., 2006, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA.

[28] Tao, S., Shintani, K., Bostanabad, R., Chan, Y. C., Yang, G., Meingast, H., and Chen, W., 2017, "Enhanced Gaussian Process Metamodeling and Collaborative Optimization for Vehicle Suspension Design Optimization," ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, Aug. 6–9.

[29] Zhang, Y., Tao, S., Chen, W., and Apley, D. W., 2019, "A Latent Variable Approach to Gaussian Process Modeling With Qualitative and Quantitative Factors," Technometrics, **62**(3), pp. 291–302.

[30] Wang, Y., Iyer, A., Chen, W., and Rondinelli, J. M., 2020, "Featureless Adaptive Optimization Accelerates Functional Electronic Materials Design," Appl. Phys. Rev., **7**(4), p. 041403.

[31] Qian, P. Z. G., Wu, H., and Wu, C. F. J., 2008, "Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors," Technometrics, **50**(3), pp. 383–396.

[32] Deng, X., Lin, C. D., Liu, K.-W., and Rowe, R. K., 2017, "Additive Gaussian Process for Computer Models With Qualitative and Quantitative Factors," Technometrics, **59**(3), pp. 283–292.

[33] Oune, N., and Bostanabad, R., 2021, "Latent Map Gaussian Processes for Mixed Variable Metamodeling," Comput. Methods Appl. Mech. Eng., **387**(C), p. 114128.

[34] Gallager, R. G., 2013, *Stochastic Processes: Theory for Applications*, Cambridge University Press, Cambridge, UK.

[35] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Trans. Evol. Comput., **6**(2), pp. 182–197.
[36] Toal, D. J. J., Bressloff, N. W., Keane, A. J., and Holden, C. M. E., 2011, "The Development of a Hybridized Particle Swarm for Kriging Hyperparameter Tuning," Eng. Optim., **43**(6), pp. 675–699.
[37] Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., 1997, "Algorithm 778: L-BFGS-B," ACM Trans. Math. Softw., **23**(4), pp. 550–560.
[38] Bostanabad, R., Chan, Y.-C., Wang, L., Zhu, P., and Chen, W., 2019, "Globally Approximate Gaussian Processes for Big Data With Application to Data-Driven Metamaterials Design," ASME J. Mech. Des., **141**(11), p. 111402. .
[39] Tripathy, R., Bilionis, I., and Gonzalez, M., 2016, "Gaussian Processes With Built-in Dimensionality Reduction: Applications to High-Dimensional Uncertainty Propagation," J. Comput. Phys., **321**, pp. 191–223.
[40] Gardner, J. R., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G., 2018, "GPyTorch: Blackbox Matrix–Matrix Gaussian Process Inference with GPU Acceleration," 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada, Dec. 2–8.
[41] Susiluoto, J., Spantini, A., Haario, H., Härkönen, T., and Marzouk, Y., 2020, "Efficient Multi-Scale Gaussian Process Regression for Massive Remote Sensing Data With satGP v0.1.2," Geosci. Model Dev., **13**(7), pp. 3439–3463.
[42] Stanton, S., Maddox, W., Delbridge, I., and Wilson, A. G., 2021, "Kernel Interpolation for Scalable Online Gaussian Processes," Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, B. Arindam and F. Kenji, eds., Online, Apr. 13–15, pp. 3133–3141.
[43] Planas, R., Oune, N., and Bostanabad, R., 2020, "Extrapolation With Gaussian Random Processes and Evolutionary Programming," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Online, Aug. 17–19, American Society of Mechanical Engineers, p. V11AT11A004.
[44] Planas, R., Oune, N., and Bostanabad, R., 2021, "Evolutionary Gaussian Processes," ASME J. Mech. Des., **143**(11), p. 111703.
[45] Moon, H., 2010, *Design and Analysis of Computer Experiments for Screening Input Variables*, The Ohio State University, Columbus, OH.
[46] Morris, M. D., Mitchell, T. J., and Ylvisaker, D., 1993, "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," Technometrics, **35**(3), pp. 243–255.
[47] Tuo, R., and Wu, C. F. J., 2015, "Efficient Calibration for Imperfect Computer Models," Ann. Stat., **43**(6), pp. 2331–2352.
[48] Tuo, R., and Jeff Wu, C. F., 2016, "A Theoretical Framework for Calibration in Computer Models: Parametrization, Estimation and Convergence Properties," SIAM/ASA J. Uncertain. Quantif., **4**(1), pp. 767–795.
[49] Park, C., and Apley, D., 2018, "Patchwork Kriging for Large-Scale Gaussian Process Regression," J. Mach. Learn. Res., **19**(1), pp. 269–311.