Modelling uncertainty in experimental and clinical measurements

Multi-fidelity Gaussian Process

VITAL Training School 02

February 11, 2026

VITAL

Introduction

The Challenge

- **High-Fidelity Data - $f_{high}(\boldsymbol{x})$:**
  - → Expensive to acquire (e.g., 3D electromechanical simulation, 3D CFD simulations, 1D arterial models, clinical trials, experimental tests).
  - → Sparse/limited quantity.
  - → "Ground Truth".
- **Low-Fidelity Data - $f_{low}(\boldsymbol{x})$:**
  - → Cheap to acquire (e.g., 0D models, simplified physics, lower fidelity experiments).
  - → Abundant.
  - → Biased or approximate.

Goal of today's workshop:

Combine both to make accurate predictions with limited budget.

VITAL

Background

INTERFACE

rsif.royalsocietypublishing.org

Research

Cite this article: Perdikaris P, Karniadakis GE. 2016 Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. *J. R. Soc. Interface* **13**: 20151107. http://dx.doi.org/10.1098/rsif.2015.1107

Received: 24 December 2015
Accepted: 21 April 2016

Subject Category:
Life Sciences – Mathematics interface

Subject Areas:
bioengineering, biomathematics, biophysics

Keywords:

Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond

Paris Perdikaris[1] and George Em Karniadakis[2]

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

We present a computational framework for model inversion based on multi-fidelity information fusion and Bayesian optimization. The proposed methodology targets the accurate construction of response surfaces in parameter space, and the efficient pursuit to identify global optima while keeping the number of expensive function evaluations at a minimum. We train families of correlated surrogates on available data using Gaussian processes and auto-regressive stochastic schemes, and exploit the resulting predictive posterior distributions within a Bayesian optimization setting. This enables a smart adaptive sampling procedure that uses the predictive posterior variance to balance the exploration versus exploitation trade-off, and is a key enabler for practical computations under limited budgets. The effectiveness of the proposed framework is tested on three parameter estimation problems. The first two involve the calibration of outflow boundary conditions of blood flow simulations in arterial bifurcations using multi-fidelity realizations of one- and three-dimensional models, whereas the last one aims to identify the forcing term that generated a particular solution to an elliptic partial differential equation.

multi-fidelity modelling, Bayesian optimization, inverse problems, blood flow simulations, outflow conditions, machine learning

Author for correspondence:
Paris Perdikaris
e-mail: parisp@mit.edu

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

Inverse problems are ubiquitous in science. Being inherently ill-posed, they require solution paths that often challenge the limits of our understanding, as reflected by our modelling and computing capabilities. Unlike forward problems, in inverse problems we have to numerically solve the principal equations (i.e. the forward problem) multiple times, often hundreds of times. The complexity in repeatedly solving the forward problem is further amplified in the presence of nonlinearity, high-dimensional parameter spaces and massive datasets; all common features in realistic physical and biological systems. The natural setting for model inversion finds itself within the principles of Bayesian statistics, which provides a formal ground for parameter estimation, i.e. the process of passing from prior belief to a posterior predictive distribution in view of data. Here, we leverage recent advances in machine learning algorithms for high-dimensional input spaces and big data [1] to design a framework for parameter estimation in blood flow models of the human circulation. The proposed paradigm aims at seamlessly using all available information sources, e.g. experimental measurements, computer simulations, empirical laws, etc., through a general multi-fidelity information fusion methodology in which surrogate models are trained on available data, enabling one to explore the interplay between all such sources in a systematic manner.

In general, we model the response of a system as a function $y = g(\mathbf{x})$ of d input parameters $\mathbf{x} \in \mathbb{R}^d$. The goal of model inversion is to identify the parametric configuration in $\mathbf{x}$ that matches a target response y^*. This translates into solving the following optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \|g(\mathbf{x}) - y^*\|, \tag{1.1}$$

Perdikaris, Karniadakis — 2016

Open full paper

- Perdikaris, Karniadakis — 2016

- Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond.
- Provides some background on today's workshop, although we have made some changes to the original algorithm to make it more efficient and user-friendly.

What is a Gaussian Process?

Before mixing data sources, we must understand the core tool: **Gaussian Processes (GP)**.
A GP is a **distribution over functions**. Instead of predicting a single value y for an input x, it predicts a **Normal distribution** of possible values.

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$$

- **Mean Function** $\mu(x)$**:** The "average" or expected shape of the function.
- **Covariance Kernel** $k(x, x')$**:** Describes how "similar" two points are. If x and x' are close, $f(x)$ and $f(x')$ should be similar.

Why use GPs? Uncertainty!

The superpower of GPs is **Uncertainty Quantification**.

1. **Interpolation:** At observed data points, the uncertainty (variance) collapses to zero (or noise level).
2. **Extrapolation:** Far from data, the uncertainty grows, admitting “we don’t know”.

This makes GPs ideal for experimental design and optimization—they tell us *where* we need more data.

Theory: Kennedy & O'Hagan (2000)

Now, how do we combine *two* sources of data? The seminal framework uses an **Autoregressive Formulation**.
We model the high-fidelity function as a scaled version of the low-fidelity function plus a bias correction:

$$f_{high}(x) = \rho \cdot f_{low}(x) + \delta(x)$$

- ρ: Constant scaling factor (correlation between fidelities).
- $f_{low}(x)$: The low-fidelity function (modeled as a GP).
- $\delta(x)$: The **Bias Term** (difference between high and low fidelity), modeled as an independent GP.

Intuition: "The high-fidelity model is just the low-fidelity model, shifted and scaled, plus some error correction."

Recursive Formulation

This concept naturally extends to M levels of fidelity (e.g., 0D $\rightarrow$ 1D $\rightarrow$ 3D models).

$$f_t(x) = \rho_{t-1} f_{t-1}(x) + \delta_t(x)$$

- **Information Flow:** We learn f_1 (cheapest), use it to help learn f_2, and so on up to f_M.
- **Efficiency:** We need fewer high-fidelity points because we are only learning the *difference* (δ_t), which is often simpler than the full function f_t.

Limitations of Kennedy O'Hagan

While powerful, the original formulation has challenges:

1. **Inference Cost:** Coupled GPs can be expensive to train ($O(N^3)$), where N is total points.
2. **Assumption:** Assumes strictly hierarchical data (nested information).

Modern Approach: BoTorch

BoTorch (and GPyTorch) generalizes this using an **Index Kernel** approach. Instead of separate GPs linked by an equation, we treat "Fidelity" as just another **input dimension**.

$$x \in \mathbb{R}^d \quad \longrightarrow \quad \tilde{x} = (x, s) \in \mathbb{R}^{d+1}$$

- x: Design parameters (e.g., stiffness, pressure).
- s: Fidelity index (e.g., 0 for low-fi, 1 for high-fi).

The Kernel Structure

The covariance kernel becomes a product of a kernel over space x and a kernel over fidelity s:

$$k((x,s),(x',s')) = k_{input}(x,x') \otimes k_{fidelity}(s,s')$$

- k_{input}: Standard Matern or RBF kernel. "Points close in space are correlated."
- $k_{fidelity}$: Index Kernel. "Points from different fidelities are correlated based on the task similarity."

This allows the model to learn that "Low-Fi at x" provides information about "High-Fi at x".

SingleTaskMultiFidelityGP

In BoTorch, the `SingleTaskMultiFidelityGP` class handles this structure automatically.

1. **Input Augmentation:** It appends the fidelity index column to your data.
2. **Linear Truncated Kernel:** A specific kernel choice that mimics the Kennedy & O'Hagan hierarchy (High depends on Low, but not vice versa).
3. **Efficiency:** It exploits the structure for faster mathematical operations (inference).

Example: The Forrester Function

Let's visualize the problem.

- **True Function (Right):** Complex, high-frequency, "High Fidelity".
- **Low Fidelity (Left):** Captures the general trend, but is shifted and scaled. This is our "cheap approximation".

Optimization Goal

We want to find the global optimum (e.g., maximum) of the **High-Fidelity** function $f_{high}(x)$.

$$x^* = \arg\max_x f_{high}(x)$$

Since evaluating f_{high} is expensive (e.g., days of computation), we cannot just sweep the parameter space. Instead, we use **Bayesian Optimization** guided by our Multi-Fidelity GP.

Acquisition Function: Expected Improvement

To decide where to sample next, we use an **Acquisition Function**. A popular choice is **Expected Improvement (EI)**. It asks: *"How much do we expect to improve over our current best result f^* if we sample at x?"*
Or, in other words for our specific optimization problem: *"How much do we expect $f(x)$ to be higher than our current highest f^* value if we sample at x?"*,

$$\alpha_{EI}(x) = \mathbb{E}[\max(f(x) - f^*, 0)]$$

- If the model predicts $f(x)$ is likely worse than f^*, EI is low.
- If $f(x)$ is likely better, EI is high.

The Exploration-Exploitation Trade-off

EI automatically balances two competing goals using the GP's predictions ($\mu(x), \sigma(x)$):

1. **Exploitation:** Sampling where the predicted mean $\mu(x)$ is high. "Go where we think it's good."
   - Term: $(\mu(x) - f^*)$
2. **Exploration:** Sampling where the uncertainty $\sigma(x)$ is high. "Go where we don't know anything."
   - Term: $\sigma(x)$

Result: We don't get stuck in local optima (thanks to exploration) but we converge quickly (thanks to exploitation).

EI in Multi-Fidelity Context

In our Multi-Fidelity setup, we are only interested in optimizing the high-fidelity response.

- We calculate EI on the **High-Fidelity** slice of the model ($s = 1.0$).
- The **Low-Fidelity** data plays a crucial role: it reduces the uncertainty $\sigma(x)$ globally via the learned correlations.
- This effectively "guides" the acquisition function to focus high-fidelity samples only in the most promising regions, ignoring areas where the low-fidelity model already says "this is bad".

Practice 1: botorch Introduction

Open the Google Colab notebook `vital26TS02_botorch_multifidelity_1D.ipynb` to see this in action.
We will:

1. Fit a Multi-Fidelity GP to the Forrester function data.
2. Visualize the acquisition function (EI) and how it evolves as we add more high-fidelity samples.

Circulatory Model Tuning

Practice 2 - PART A

Let's apply this to a complex physiological problem: **Tuning a Circulatory System Model**.

- **Problem:** Identify 5 key parameters (3 Resistances, 2 Compliances) to match 4 clinical measurements (Flows, Pressures).
- **Fidelities:**
  - → **Low-Fi:** Simplified linear model (fast).
  - → **High-Fi:** Nonlinear model with coupling (slow, accurate).
- **WARNING:** This is an example problem, with no true physiological meaning. More physiological non-linearities for the high-fidelity model, e.g. pressure-dependent compliance, stenotic restrictions, etc. can be added by the users themselves later.

The Setup

- **Parameters:** $\theta = [R_1, R_2, R_3, C_1, C_2]$
- **Measurements:** $Y = [Q_1, Q_2, P_1, P_2]$
- **Objective:** Minimize Weighted Mean Squared Error (MSE) between model output and clinical targets.

Optimization Process:

1. Initialize with 100 Low-Fi samples and 10 High-Fi samples.
2. Iteratively query High-Fi model where EI is highest.
3. Update Multi-Fidelity GP model.

Open the Google Colab notebook `vital26TS02_circulatory_model_tuning.ipynb` to see this in action.

Results: Convergence

The optimizer quickly converges to improved pressure and flow predictions.

- **Blue Line:** Objective value (MSE) decreases rapidly.
- **Red Line:** The distance to the true parameters drops, showing we are finding improved parameteric instances representing the physiological state.

Note the increase in L2 parameter error upon improving the MSE. This is a common phenomenon in inverse problems, where multiple parameter sets can produce similar outputs (non-identifiability).

Beyond Optimization: Posterior Estimation

Practice 2 - PART B

Finding the *best* parameter set is often not enough in medicine.
We need to know **how uncertain** we are.
This requires us to estimate the full posterior distribution $p(\theta|Y_{obs})$.

$$p(\theta|Y_{obs}) \propto p(Y_{obs}|\theta)p(\theta)$$

This is classically done with Markov Chain Monte Carlo (as discussed during the first VITAL training school `VITAL25TS01_Krijnen_Peirlinck`),
but this requires evaluating the likelihood $p(Y_{obs}|\theta)$ for many θ samples, which requires expensive simulations, and is thus computationally prohibitive.

Approximate Bayesian Computation

with Informed Prior

Approximate Bayesian Computation (ABC) allows us to estimate the full posterior distribution $p(\theta|Y_{obs})$.

Methodology:

1. **Likelihood-Free Inference:** We avoid explicit likelihoods by accepting samples that produce data "close enough" to observations.
2. **Informed Prior Construction:** Instead of sampling blindly, we construct a proposal distribution which estimates the probability that a given θ will produce a low error.
3. **Efficiency:** This focuses computational effort on regions the model already thinks are promising.

We actually already have our trained **Multi-Fidelity GP** which provides us insights into which parameters θ lead to a good fit, and those who don't.

Therefore, we can use our **Multi-Fidelity GP** to construct the informed prior proposal distribution.

The Informed Prior Equation

We define the proposal distribution proportional to the probability that the error is small:

$$p_{GP}(\theta) \propto \frac{1}{\sqrt{\sigma^2_{GP}(\theta) + \sigma^2_{GP}(\theta_{min})}} \exp\left(-\frac{(\mu_{GP}(\theta) - \mu_{GP}(\theta_{min}))^2}{2(\sigma^2_{GP}(\theta) + \sigma^2_{GP}(\theta_{min}))}\right)$$

- $\mu_{GP}(\theta)$: The GP's predicted error (MSE).
- $\sigma^2_{GP}(\theta)$: The GP's uncertainty about that error.
- θ_{min}: The best parameter set observed in high-fidelity training data so far (lowest error).
- $\mu_{GP}(\theta_{min})$: The GP's predicted error at the best observed θ_{min}.
- $\sigma_{GP}(\theta_{min})$: The GP's uncertainty at the best observed parameter set θ_{min}.

Interpretation: High probability is assigned to regions where the predicted error is likely to be close to or better than the current best.

The Algorithm (Rejection ABC)

1. **Generate Candidates:** Draw samples from the parameter space.
2. **Filter by Prior:** Keep sample θ_i with probability $\propto p_{GP}(\theta_i)$ (Rejection Sampling).
3. **Simulation:** Run the **High-Fidelity** model for the surviving candidates.
4. **ABC Acceptance:** Keep θ_i if the simulated error is below a threshold ϵ.

Note on the Threshold: The tolerance ϵ should be carefully chosen based on the **noise level** in the experimental or clinical data. If ϵ is too small, we might not accept any samples; if too large, we lose the ability to resolve the parameters. Open PART B of the Google Colab notebook `vital26TS02_circulatory_model_tuning.ipynb` to see this in action.

Posterior Results

The resulting posterior distribution tells us:

- **Parameter correlations:** e.g., "If R_2 is low, C_1 must be high."
- **Identifiability:** Narrow peaks mean the parameter is well-determined. Broad distributions mean many values fit the data equally well.

Clinical Value

We can report
"Resistance 1 is 1.66 ± 0.55" instead of just "1.66".
"Compliance 1 is 1.61 ± 0.78" instead of just "1.61".
...

Parameter	Best Point	Post. Mean	Post. Std	95% CI
R1	0.5885	1.6610	0.5592	[1.5768, 1.7666]
R2	0.4254	1.3494	0.5553	[1.2580, 1.4339]
R3	0.4558	1.4589	0.6017	[1.2927, 1.6091]
C1	0.4562	1.6143	0.7813	[1.2110, 2.1482]
C2	0.5612	1.6648	0.8878	[1.0869, 2.2358]

Conclusion

- **Gaussian Processes:** Powerful tools for regression with uncertainty.
- **Multi-Fidelity GPs:** Bridge the gap between cheap/inaccurate and expensive/accurate data.
- **Bayesian Optimization:** Efficiently finds optimal parameters.
- **ABC + MFGP:** Enables full uncertainty quantification at a fraction of the cost of standard MCMC.
- **Impact:** significantly reduces computational cost for optimization and uncertainty quantification in VITAL applications.

Final assignment: Pulse wave model calibration

We are modeling a simple arterial bifurcation with one parent vessel and two daughter vessels. The model simulates the pressure waveforms at the outlets given input parameters E_1 and E_2 (the Young's moduli of the daughter vessels). This simple arterial biforucation model can be modeled with two levels of fidelity:

1. **Low-Fidelity (LF):** A version taking small amount of discretization points.
2. **High-Fidelity (HF):** A version taking a finer discretization (e.g. 100 nodes per vessel).

The low-fidelity model is computationally cheap but less accurate, while the high-fidelity model is more expensive but provides a better match to reality.

We have measured the outlet pressure waveform $P_{\text{out}}(t)$ from an experiment, but these measurements are noisy. We want to calibrate our model parameters E_1 and E_2 to match these measurements as closely as possible.

Task definition

Your task: use multi-fidelity Bayesian optimization to calibrate the model parameters E_1 and E_2 to match the noisy outlet-pressure measurements $P_{\text{out}}(t)$ provided in `pulse_wave_data.npz`.

Proposed steps to complete the assignment

Complete the missing parts of the notebook (see the final "Instructions" cell), including:

1. **BoTorch imports** for multi-fidelity modeling and optimization.
2. Create multi-fidelity training data by evaluating both the LF and HF models at different parameter settings.
3. Definining an **objective function** that:
   - accepts an input tensor with a fidelity feature,
   - evaluates the corresponding model (LF or HF),
   - returns the **negative** MSE (so larger is better).
4. Initialize and fit a **multi-fidelity GP model** (e.g. `SingleTaskMultiFidelityGP`) with appropriate input/output transforms.
5. A **Bayesian optimization loop**:
   - build an acquisition function (e.g. (log-)Expected Improvement),
   - iteratively select the next candidate at **high fidelity** (fix fidelity feature to HF),

VITAL

- plot the convergence of the best MSE and the distance to the true parameters over iterations.
- optionally add **additional LF samples** per iteration (cost-aware exploration).

6. Plot the best fitted pressure waveform against the noisy measurements.

Optional extensions

If you finish early, try one of the following:

1. Add a **cost model** and compare strategies (HF-only vs multi-fidelity).
2. Add posterior plots of predicted pressure uncertainty bands around the best solution using ABC.

Francisco Sahli Costabal | Mathias Peirlinck

Speaker notes