Introduction to the calibration of computer models

Richard Wilkinson

University of Nottingham

What is calibration?

Parameter estimation/fitting/tuning/inverse problem/system identification

Computer model (or *simulator*) $f(x, u)$

- x a parameter, we want to estimate
- u a control input.

What is calibration?

Parameter estimation/fitting/tuning/inverse problem/system identification

Computer model (or *simulator*) $f(x, u)$

- x a parameter, we want to estimate
- u a control input.

Given

- observations y collected under experimental conditions u
- a relationship between $f(x^*, u)$ and y

calibration is the process of estimating x^* - the 'best input'

What is calibration?

Parameter estimation/fitting/tuning/inverse problem/system identification

Computer model (or *simulator*) $f(x, u)$

- x a parameter, we want to estimate
- u a control input.

Given

- observations y collected under experimental conditions u
- a relationship between $f(x^*, u)$ and y

calibration is the process of estimating x^* - the 'best input'

- Adjust x^* until the simulator prediction looks like the data y.

What is calibration?

Parameter estimation/fitting/tuning/inverse problem/system identification

Computer model (or *simulator*) $f(x, u)$

- x a parameter, we want to estimate
- u a control input.

Given

- observations y collected under experimental conditions u
- a relationship between $f(x^*, u)$ and y

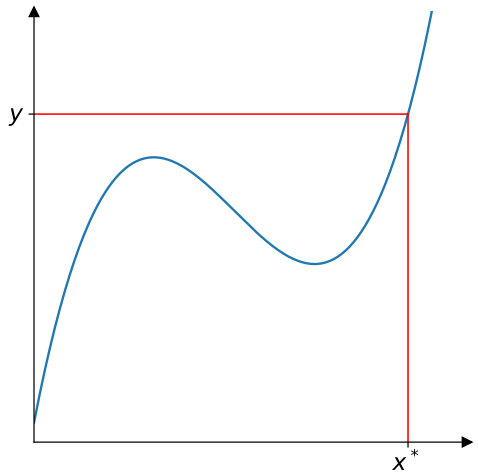
calibration is the process of estimating x^* - the 'best input'

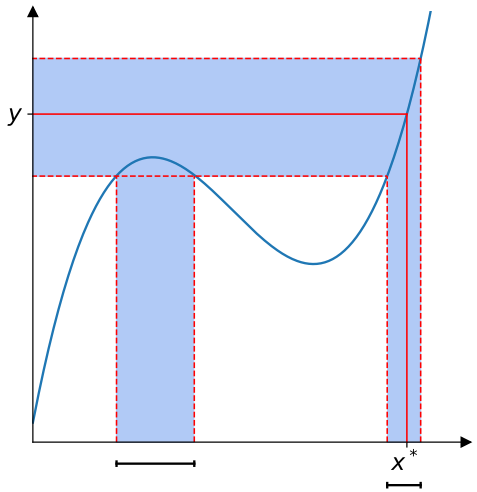
- Adjust x^* until the simulator prediction looks like the data y.

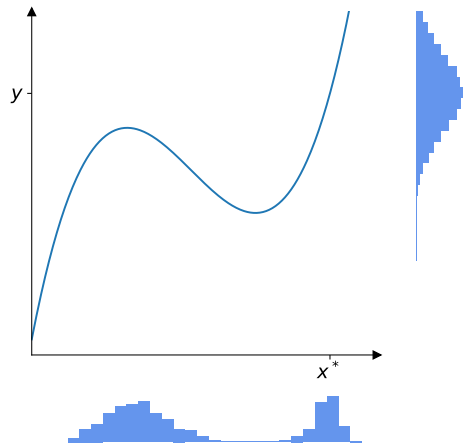
Stochastic models:

$$f(x, u, \xi) \text{ where } \xi \sim U[0, 1]^D$$

Calibration



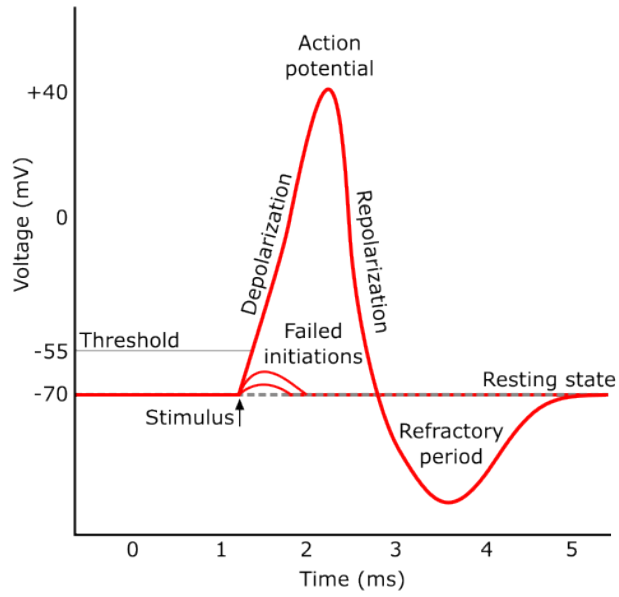




Motivating example: Cardiac Digital Twins

Motivating example: Cardiac Digital Twins

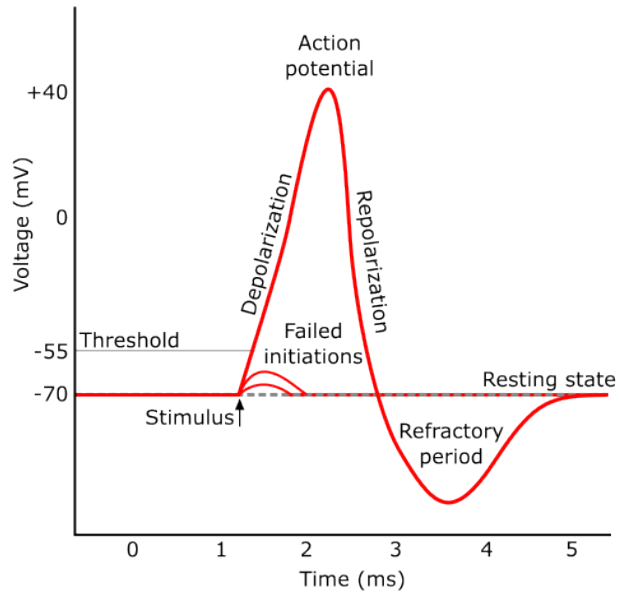
The heart is an electrical-mechanical pump, which contracts under electrical potential.



Motivating example: Cardiac Digital Twins

The heart is an electrical-mechanical pump, which contracts under electrical potential.

- Left atrium - sinus rhythm



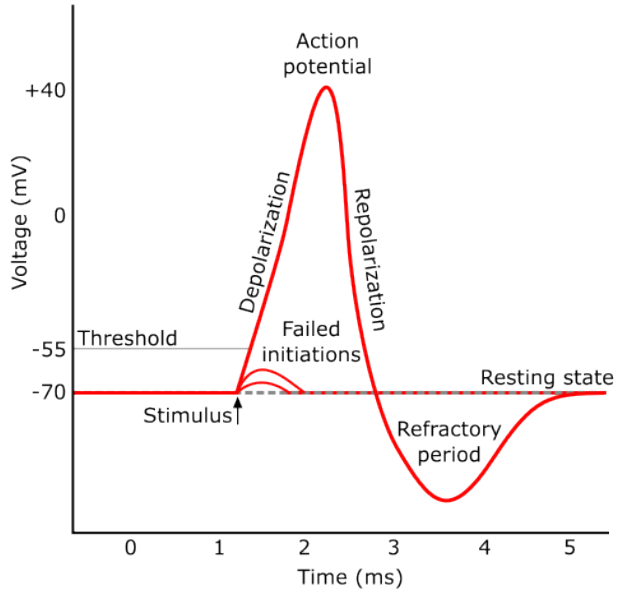
Motivating example: Cardiac Digital Twins

The heart is an electrical-mechanical pump, which contracts under electrical potential.

- Left atrium - sinus rhythm

Atrial fibrillation (AF) is rapid and uncoordinated electrical activation (arrhythmia) leading to poor mechanical function.

- Some hearts sustain AF - others don't.



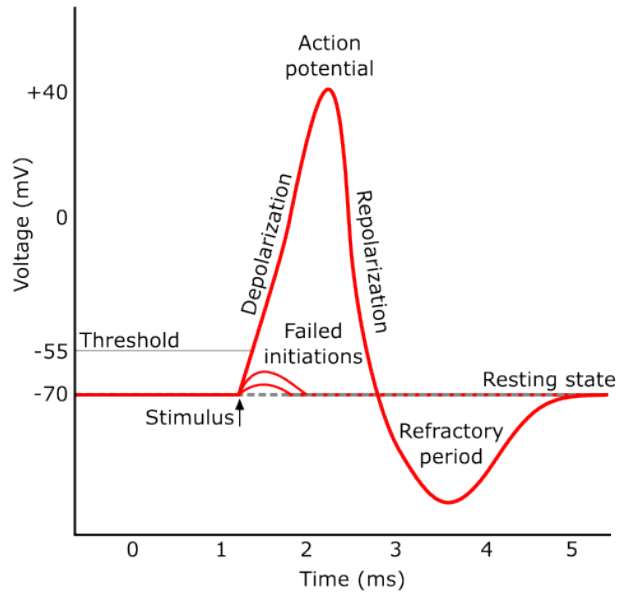
Motivating example: Cardiac Digital Twins

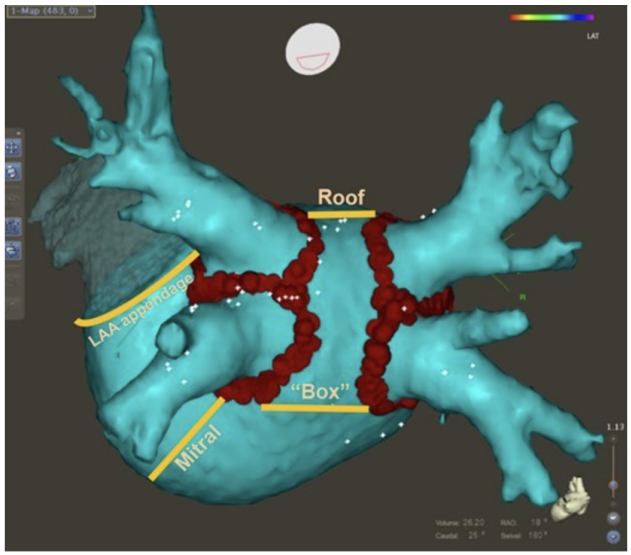
The heart is an electrical-mechanical pump, which contracts under electrical potential.

- Left atrium - sinus rhythm

Atrial fibrillation (AF) is rapid and uncoordinated electrical activation (arrhythmia) leading to poor mechanical function.

- Some hearts sustain AF - others don't.
- Affects around 600,000 people in UK.
- Catheter ablation removes/isolates pathological tissue that sustain/initiates AF.
- Treatment unsuccessful in ≈40% of patients .





Kirchof & Calkins 2017

Modelling activation

Corrado & Niederer 2016

Given an atrial geometry $\mathcal{G}$, the simulator f models the voltage through time $v \equiv v(z, t)$ where $z \in \mathcal{G}$.

Modelling activation

Corrado & Niederer 2016

Given an atrial geometry $\mathcal{G}$, the simulator f models the voltage through time $v \equiv v(z,t)$ where $z \in \mathcal{G}$.

$$\frac{\partial v}{\partial t} = \nabla \cdot (D \nabla v) + h \frac{v\,(v - v_{gate})\,(1 - v)}{\tau_{in}} - (1 - h)\frac{v}{\tau_{out}} + u_{stim}$$

$$\frac{\partial h}{\partial t} = \begin{cases} (1 - h)/\tau_{open} & \text{if } v \leq v_{gate} \\ -h/\tau_{open} & \text{otherwise} \end{cases}$$

- Parameters $x = \{\tau_{open}(z), \tau_{out}(z), \tau_{in}(z), D(z)\}$
- Control inputs $u_{stim}(z,t)$

Modelling activation

Corrado & Niederer 2016

Given an atrial geometry $\mathcal{G}$, the simulator f models the voltage through time $v \equiv v(z,t)$ where $z \in \mathcal{G}$.

$$\frac{\partial v}{\partial t} = \nabla \cdot (D \nabla v) + h \frac{v\,(v - v_{gate})\,(1 - v)}{\tau_{in}} - (1 - h)\frac{v}{\tau_{out}} + u_{stim}$$

$$\frac{\partial h}{\partial t} = \begin{cases} (1 - h)/\tau_{open} & \text{if } v \leq v_{gate} \\ -h/\tau_{open} & \text{otherwise} \end{cases}$$

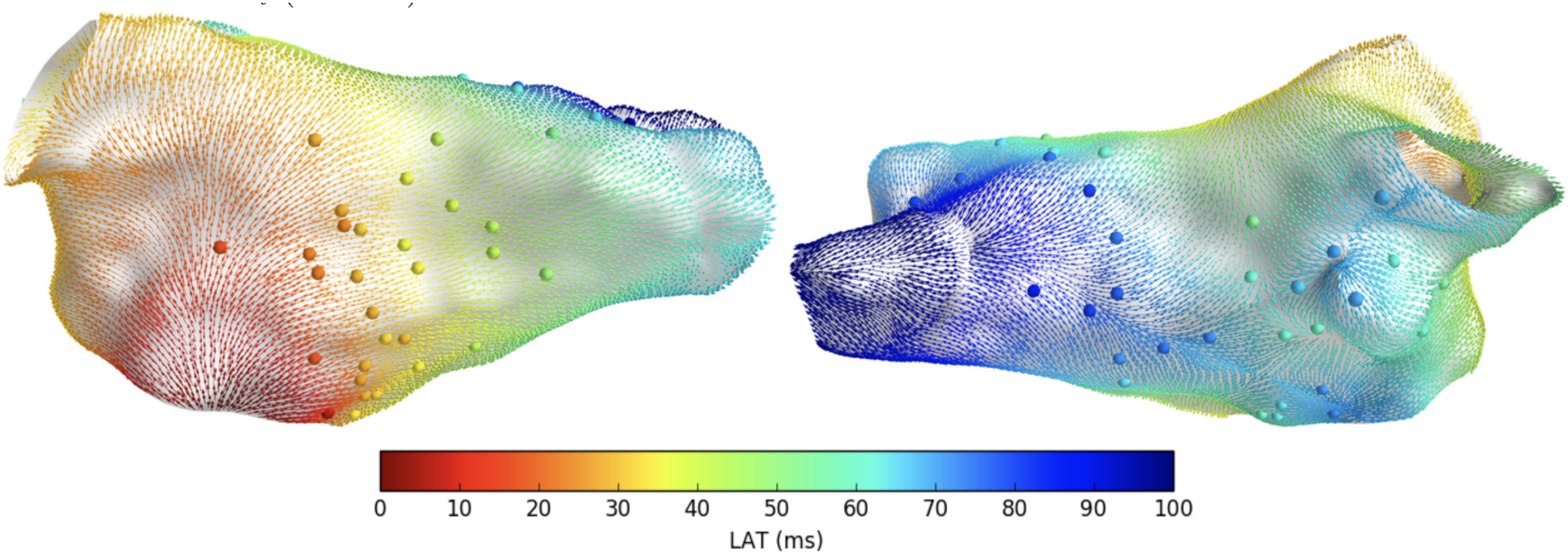
- Parameters $x = \{\tau_{open}(z), \tau_{out}(z), \tau_{in}(z), D(z)\}$
- Control inputs $u_{stim}(z,t)$

Each simulation takes $\sim$ an hour on a HPC.
Simulations are different for every patient specific geometry $\mathcal{G}$

Impossible dreams

Coveney et al. 2022

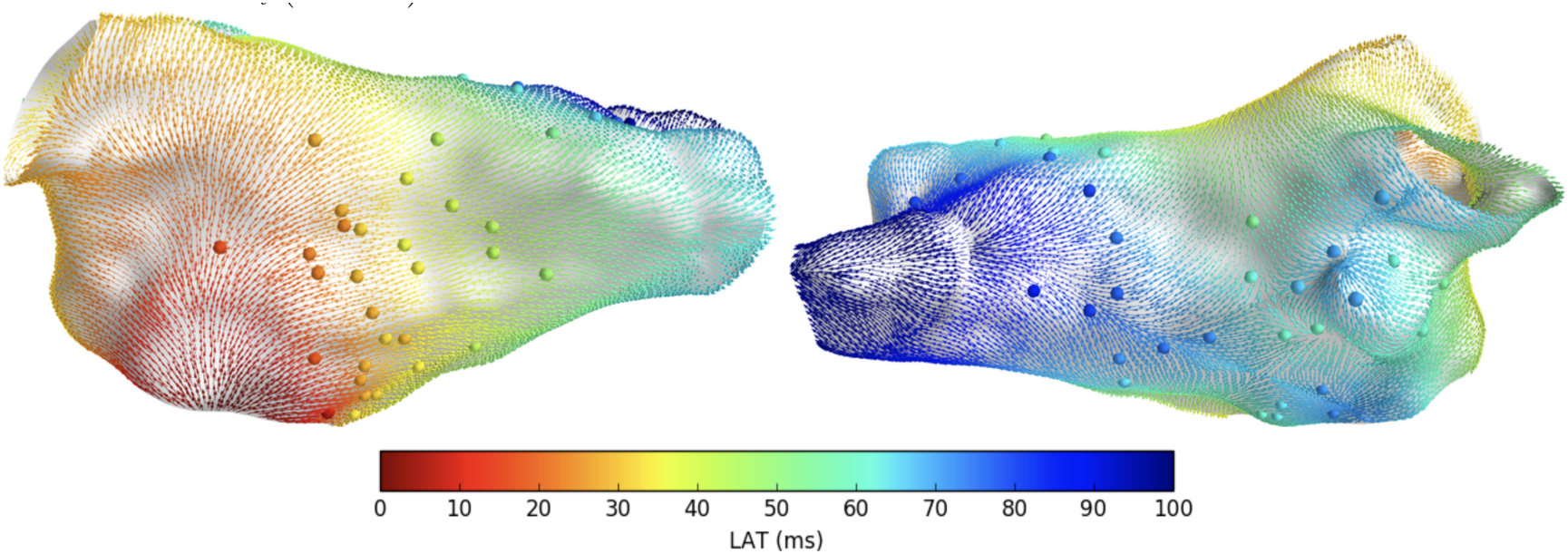
In the clinic, cardiologists pace the heart (i.e. fix u_{stim}) and collect noisy measurements of local activation times (LAT) at some locations.



Impossible dreams

Coveney et al. 2022

In the clinic, cardiologists pace the heart (i.e. fix u_{stim}) and collect noisy measurements of local activation times (LAT) at some locations.



We need to estimate parameters:

$$\pi(x^*|y) \text{ where } y = f(x^*, u = F_{stim}) + e$$

and predict if AF will be sustained after ablation a;

$$\mathbb{P}(\text{AF sustained}|a) = \int \mathbb{P}(\text{AF sustained}|x^*, a)\pi(x^*|y)\mathrm{d}x^*$$

Challenges of calibrating cardiac digital twins

Complex inference problem

- High dimensional parameter x with sparse noisy data y
- Expensive simulator f
- Uncertain geometry

Challenges of calibrating cardiac digital twins

Complex inference problem

- High dimensional parameter x with sparse noisy data y
- Expensive simulator f
- Uncertain geometry

To be a practical clinical tool

- Inference needs to be fast, cheap, and scalable
- Predictions need to be robust
- Need calibration approaches that can cope with variable data (missing, incomplete, asynchronous etc).

Key decisions in calibration

W. and Lanyon 2024

- Quantity of interest
  - what do we want to compute?
- Calibration framework
  - how do we define a good fit and characterize uncertainty?
- Observation error
  - how the simulator relates to the data?
- Computation
  - how will we compute it?

Preliminaries

Identifiability

- x is structurally non-identifiable if $f(x, u) = f(x', u)$ for some $x \neq x'$
- x is practically identifiable if $\mathcal{I}(x) = \mathbb{E}\nabla \log p(y|x) \nabla \log p(y|x)^\top$ is full rank

Does the non-identifiability matter for your prediction task?

Preliminaries

Identifiability

- x is structurally non-identifiable if $f(x, u) = f(x', u)$ for some $x \neq x'$
- x is practically identifiable if $\mathcal{I}(x) = \mathbb{E}\nabla \log p(y|x) \nabla \log p(y|x)^\top$ is full rank

Does the non-identifiability matter for your prediction task?
We often use sensitivity as a proxy for identifiability.

- Sobol indices

$$S_i = \frac{\mathbb{V}\text{ar}_{X_i}(\mathbb{E}_{X_{-i}}(Y|X_i))}{\mathbb{V}\text{ar}(Y)} \qquad S_{T_i} = \frac{\mathbb{E}_{X_{-i}}(\mathbb{V}\text{ar}_{X_i}(Y|X_{-i}))}{\mathbb{V}\text{ar}(Y)}$$

  $S_i, S_{T_i} = 0$ suggests x_i is not identifiable
  - Fix non-identifiable params to a reference value

Why calibrate?

Quantity of interest (QoI)

Why calibrate?

Quantity of interest (QoI)

Explanation

- can we make our simulator output look like the true DGP?
- model development - what do we need to fix to achieve this?

Inference

Calibrated prediction

Decision

Why calibrate?

Quantity of interest (QoI)

Explanation

- can we make our simulator output look like the true DGP?
- model development - what do we need to fix to achieve this?

Inference $\pi(x^*|y)$:

- learn the true physical or 'best' value of some parameter.

Calibrated prediction

Decision

Why calibrate?

Quantity of interest (QoI)

Explanation

- can we make our simulator output look like the true DGP?
- model development - what do we need to fix to achieve this?

Inference $\pi(x^*|y)$:

- learn the true physical or 'best' value of some parameter.

Calibrated prediction

$$y' = f(x^*, u') \qquad \pi(y'|y) = \int \pi(f(x^*, u')|x^*)\pi(x^*|y)\mathrm{d}x^*$$

- Predict some future or unobserved quantity given the data available

Decision

Why calibrate?

Quantity of interest (QoI)

Explanation

- can we make our simulator output look like the true DGP?
- model development - what do we need to fix to achieve this?

Inference $\pi(x^*|y)$:

- learn the true physical or 'best' value of some parameter.

Calibrated prediction

$$y' = f(x^*, u') \qquad \pi(y'|y) = \int \pi(f(x^*, u')|x^*)\pi(x^*|y)\mathrm{d}x^*$$

- Predict some future or unobserved quantity given the data available

Decision

$$\max_{a \in \mathcal{A}} \mathbb{E}(\text{utility of action a}) = \int U(a, y')\pi(y'|y)\mathrm{d}x^*$$

- Take an optimal action given current state of knowledge.

Observation model

How does best simulator prediction relate to the observations? i.e.,

$$\pi(y|f(x^*, u))$$

- Often trickiest aspect of inference

Observation model

How does best simulator prediction relate to the observations? i.e.,

$$\pi(y|f(x^*, u))$$

- Often trickiest aspect of inference

For well-specified simulator, need only think about observation error

$$y = f(x^*, u) + e$$

where, eg, e_i are assumed to be iid $N(0, \sigma^2)$

Observation model

How does best simulator prediction relate to the observations? i.e.,

$$\pi(y|f(x^*, u))$$

- Often trickiest aspect of inference

For well-specified simulator, need only think about observation error

$$y = f(x^*, u) + e$$

where, eg, e_i are assumed to be iid $N(0, \sigma^2)$

From the observation error we get the link between x^* and y which allows us to learn. Eg a likelihood

$$\pi(y|x^*) = \mathcal{N}(y; f(x^*, u), \sigma^2 I)$$

Observation model

How does best simulator prediction relate to the observations? i.e.,

$$\pi(y|f(x^*, u))$$

- Often trickiest aspect of inference

For well-specified simulator, need only think about observation error

$$y = f(x^*, u) + e$$

where, eg, e_i are assumed to be iid $N(0, \sigma^2)$

From the observation error we get the link between x^* and y which allows us to learn. Eg a likelihood

$$\pi(y|x^*) = \mathcal{N}(y; f(x^*, u), \sigma^2 I)$$

Need to work harder for stochastic models

$$\pi(y|x^*) = \int \pi(y|f(x^*, u, \xi))\pi(\xi)df$$

Misspecified models

Bochkina 2022

What if the simulator is an imperfect representation of reality $\zeta(u)$?

Misspecified models

Bochkina 2022

What if the simulator is an imperfect representation of reality $\zeta(u)$?

- Kennedy and O'Hagan 2001: model the discrepancy between best simulator prediction and reality

$$\zeta(u) = f(x^*, u) + \delta(u)$$
$$y = \zeta(u) + e$$

- Typically model $\delta(u)$ as a Gaussian process (GP)
  - Can we model our way out of trouble by expanding $\mathcal{F}$ into a non-parametric world?

An appealing, but flawed, idea

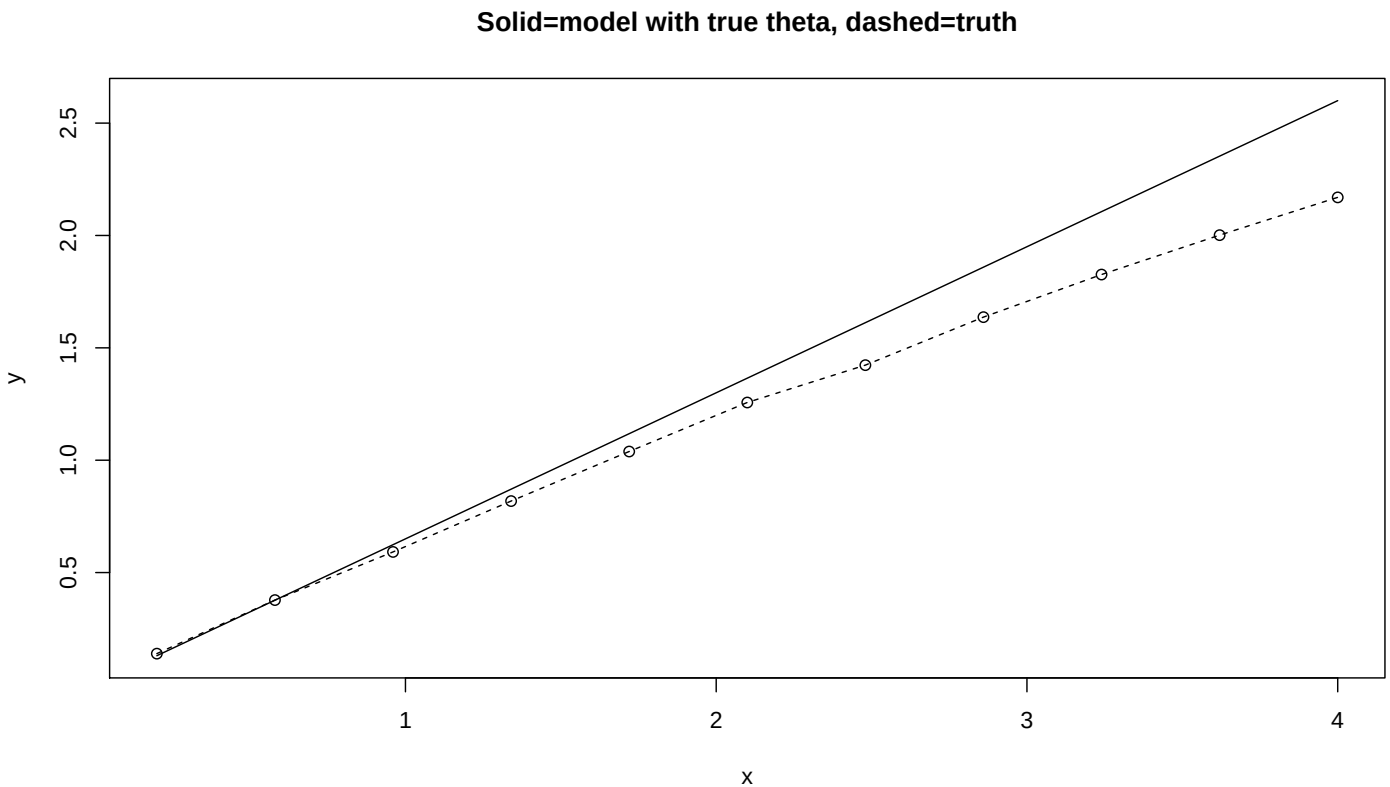
Brynjarsdottir and O'Hagan 2014

Simulator

$$f_x(u) = xu$$

Reality

$$g(u) = \frac{x^* u}{1 + \frac{u}{a}} \quad x^* = 0.66, a = 20$$

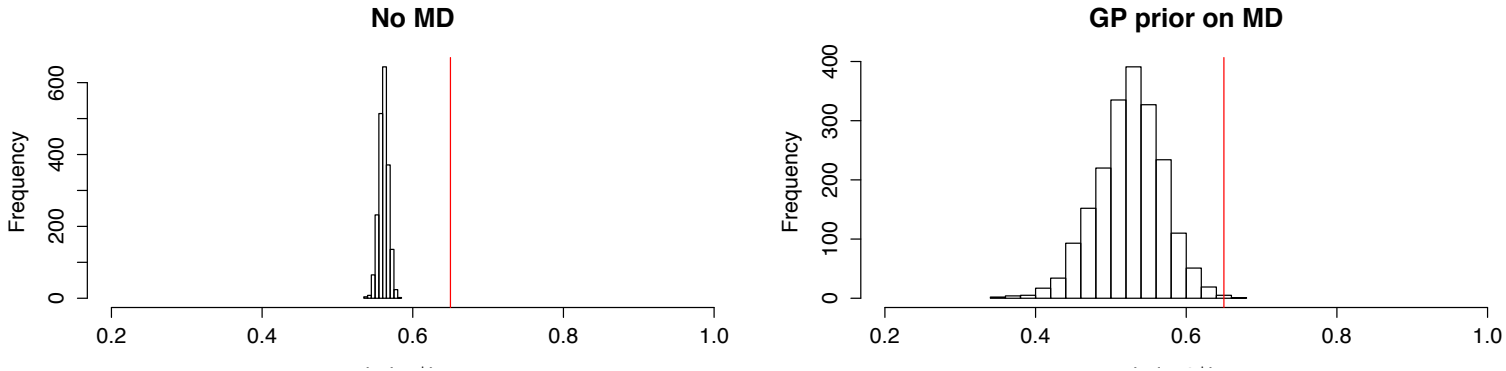


An appealing, but flawed, idea

A GP discrepancy can correct your predictions, but won't necessarily fix inference of x^* as $\delta(\cdot), x^*$ are structurally non-identifiable:

- No discrepancy:
$$y = f(x^*, u) + N(0, \sigma^2)$$

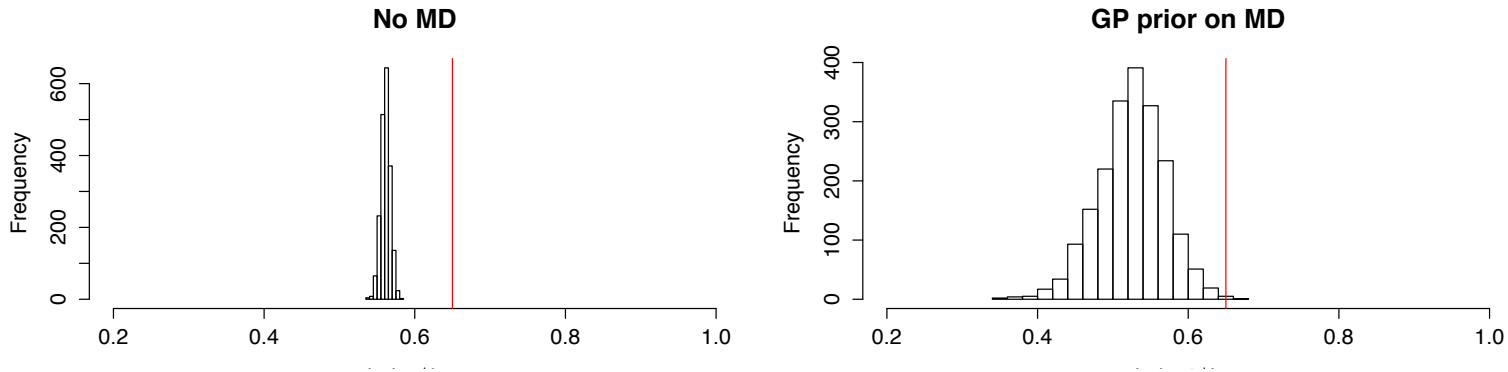
- GP discrepancy:
$$y = f(x^*, u) + \delta(u) + N(0, \sigma^2),$$
$$\delta(\cdot) \sim GP(\cdot, \cdot)$$



An appealing, but flawed, idea

A GP discrepancy can correct your predictions, but won't necessarily fix inference of x^* as $\delta(\cdot), x^*$ are structurally non-identifiable:

- No discrepancy:
$$y = f(x^*, u) + N(0, \sigma^2)$$
- GP discrepancy:
$$y = f(x^*,u) + \delta(u) + N(0, \sigma^2),$$
$$\delta(\cdot) \sim GP(\cdot, \cdot)$$



'dangerous to interpret estimates of x^*... as estimates of the true physical values of those parameters'

No explicit definition of x^* provided

Can we recover identifiability?

Sung and Tuo 2023

Let $e(x,u) = \zeta(u) - f(x,u)$
Tuo and Wu 2015 (and others) explicitly define the L_2 *projection*

$$x^* := \arg\min_x ||e(x,\cdot)||_{L_2} = \arg\min_x \int (\zeta(u) - f(x,u))^2 \mathrm{d}u$$

Can we recover identifiability?

Sung and Tuo 2023

Let $e(x, u) = \zeta(u) - f(x, u)$
Tuo and Wu 2015 (and others) explicitly define the L_2 *projection*

$$x^* := \arg\min_x ||e(x, \cdot)||_{L_2} = \arg\min_x \int (\zeta(u) - f(x, u))^2 \mathrm{d}u$$

Tuo and Wu 2016, 2020 show that if $\delta(u) \sim GP(0, k(u, u'))$, the posterior mode for x^* converges to

$$\arg\min_x ||e(x, \cdot)||_{\mathcal{H}_k} \neq x^*$$

i.e. we don't forget the prior.

Can we recover identifiability?

Sung and Tuo 2023

Let $e(x,u) = \zeta(u) - f(x,u)$
Tuo and Wu 2015 (and others) explicitly define the L_2 *projection*

$$x^* := \arg\min_x ||e(x,\cdot)||_{L_2} = \arg\min_x \int (\zeta(u) - f(x,u))^2 \mathrm{d}u$$

Tuo and Wu 2016, 2020 show that if $\delta(u) \sim GP(0, k(u,u'))$, the posterior mode for x^* converges to

$$\arg\min_x ||e(x,\cdot)||_{\mathcal{H}_k} \neq x^*$$

i.e. we don't forget the prior.

Tuo and Wu 2015 (and others) proposed frequentist approaches for estimating L_2 projection.

Can we recover identifiability?

Plumlee 2017 suggested a Bayesian approach that avoids non-identifiabilities:

- make the discrepancy $\delta(u)$ orthogonal to $\nabla_x f(x, u)$

Idea: δ shouldn't explain variations in the output that could be explained by adjusting x, ie, we shouldn't correct errors the computer model can fix

Can we recover identifiability?

Plumlee 2017 suggested a Bayesian approach that avoids non-identifiabilities:

- make the discrepancy $\delta(u)$ orthogonal to $\nabla_x f(x, u)$

Idea: δ shouldn't explain variations in the output that could be explained by adjusting x, ie, we shouldn't correct errors the computer model can fix

- Challenging to implement.

Can we recover identifiability?

Plumlee 2017 suggested a Bayesian approach that avoids non-identifiabilities:

- make the discrepancy $\delta(u)$ orthogonal to $\nabla_x f(x, u)$

Idea: δ shouldn't explain variations in the output that could be explained by adjusting x, ie, we shouldn't correct errors the computer model can fix

- Challenging to implement.

Summary:

- inference of true physical parameter values with a misspecified model is hard!
- Calibrated prediction is easier.

'dangerous to interpret estimates of x^*... as estimates of the true physical values of those parameters'

Physics informed machine learning

Raissi *et al.* 2019, Ramsey *et al.* 2007, Brunton *et al.* 2016

Suppose f solves differential equations

$$\mathcal{L}g = 0$$

E.g.

$$g_x(z,t) \text{ solution to } \partial_t g = x\nabla_z^2 g$$

Physics informed machine learning

Raissi *et al.* 2019, Ramsey *et al.* 2007, Brunton *et al.* 2016

Suppose f solves differential equations

$$\mathcal{L}g = 0$$

E.g.

$$g_x(z,t) \text{ solution to } \partial_t g = x\nabla_z^2 g$$

Physics informed neural networks fit model $h_{\psi,x}(z,t)$

$$\min_{\psi} ||\mathcal{L}_x h_{\psi,x}|| + ||h_{\psi,x}(z,t) - y||$$

Physics informed machine learning

Raissi *et al.* 2019, Ramsey *et al.* 2007, Brunton *et al.* 2016

Suppose f solves differential equations

$$\mathcal{L}g = 0$$

E.g.

$$g_x(z,t) \text{ solution to } \partial_t g = x\nabla_z^2 g$$

Physics informed neural networks fit model $h_{\psi,x}(z,t)$

$$\min_{\psi} ||\mathcal{L}_x h_{\psi,x}|| + ||h_{\psi,x}(z,t) - y||$$

Or we can seek to learn missing physics

$$\mathcal{L}g + \delta = 0$$

Physics informed machine learning

Raissi *et al.* 2019, Ramsey *et al.* 2007, Brunton *et al.* 2016

Suppose f solves differential equations

$$\mathcal{L}g = 0$$

E.g.

$$g_x(z,t) \text{ solution to } \partial_t g = x\nabla_z^2 g$$

Physics informed neural networks fit model $h_{\psi,x}(z,t)$

$$\min_{\psi} ||\mathcal{L}_x h_{\psi,x}|| + ||h_{\psi,x}(z,t) - y||$$

Or we can seek to learn missing physics

$$\mathcal{L}g + \delta = 0$$

Hard to maintain any interpretation for x.

Implicit observation models

W. 2013

Often we don't give an explicit statistical model relating observations to best simulator prediction.

- ABC/history matching, kernel methods, scoring rule approaches, Bayes linear etc

Implicit observation models

W. 2013

Often we don't give an explicit statistical model relating observations to best simulator prediction.

- ABC/history matching, kernel methods, scoring rule approaches, Bayes linear etc

But sometimes we can reinterpret the algorithm within a statistical framework. E.g.

ABC

- Draw x from $\pi(x)$
- Simulate $Y^{sim} \sim f(x, u, \xi)$
- Accept x if $d(Y^{sim}, y) \leq \epsilon$

is equivalent to assuming $\delta(u) \sim U\{e : d(y + e, y) \leq \epsilon\}$

Calibration framework

How will we characterize uncertainty about x^*?

[1] typically robust only to mild misspecification, not the typical gross misspec.

Calibration framework

How will we characterize uncertainty about x^*?

Bayes

- Given a prior distribution $\pi(x^*)$ compute posterior

$$\pi(x^*|y) = \frac{\pi(y|x^*)\pi(x^*)}{\pi(y)}$$

- More challenging with stochastic simulator
  - likelihood-free / simulation-based inference / ABC
- Often get lingering prior dependence. Calibration (cf Johanna's talk), frequentist, robustness properties?

[1] typically robust only to mild misspecification, not the typical gross misspec.

Calibration framework

How will we characterize uncertainty about x^*?

Bayes

- Given a prior distribution $\pi(x^*)$ compute posterior

$$\pi(x^*|y) = \frac{\pi(y|x^*)\pi(x^*)}{\pi(y)}$$

- More challenging with stochastic simulator
  - likelihood-free / simulation-based inference / ABC
- Often get lingering prior dependence. Calibration (cf Johanna's talk), frequentist, robustness properties?

Generalized Bayes

$$\pi(x^*|y) = \arg\min_{p \in \mathcal{P}} \mathbb{E}_{x \sim p(x)} \ell(x, y) + D(p||\pi)$$

- Generalizes Bayesian inference and allows us to use more robust[1] loss functions (see Knoblauch *et al.* 2019)
- Computation?

[1] typically robust only to mild misspecification, not the typical gross misspec.

Calibration framework

Craig *et al.* 1999, Vernon *et al.* 2014, Holden *et al.* 2018

History matching seeks to find a not ruled out yet (NROY) set

$$\mathcal{P}_{x^*} = \{x : S_{HM}(\hat{F}_x, y) \leq 3\}$$

where

$$S_{HM}(F_x, y) = \frac{|\mathbb{E}_{F_x}(Y) - y|}{\sqrt{\mathbb{V}\text{ar}_{F_x}(Y)}}$$

[2] originally $S(\hat{F}_x, y) = d(\eta(y), \eta(y'))$ where $y' \sim F_x$, or MMD etc

Calibration framework

Craig *et al.* 1999, Vernon *et al.* 2014, Holden *et al.* 2018

History matching seeks to find a not ruled out yet (NROY) set

$$\mathcal{P}_{x^*} = \{x : S_{HM}(\hat{F}_x, y) \leq 3\}$$

where

$$S_{HM}(F_x, y) = \frac{|\mathbb{E}_{F_x}(Y) - y|}{\sqrt{\mathbb{V}\text{ar}_{F_x}(Y)}}$$

ABC approximates the posterior as

$$\pi_\epsilon(x^*) \propto \pi(x^*)\mathbb{E}(\mathbb{I}_{S(\hat{F}_{x^*}, y) \leq \epsilon})$$

for some choice[2] of S and ϵ.

[2] originally $S(\hat{F}_x, y) = d(\eta(y), \eta(y'))$ where $y' \sim F_x$, or MMD etc

Calibration framework

Craig *et al.* 1999, Vernon *et al.* 2014, Holden *et al.* 2018

History matching seeks to find a not ruled out yet (NROY) set

$$\mathcal{P}_{x^*} = \{x : S_{HM}(\hat{F}_x, y) \leq 3\}$$

where

$$S_{HM}(F_x, y) = \frac{|\mathbb{E}_{F_x}(Y) - y|}{\sqrt{\mathbb{V}\text{ar}_{F_x}(Y)}}$$

ABC approximates the posterior as

$$\pi_\epsilon(x^*) \propto \pi(x^*)\mathbb{E}(\mathbb{I}_{S(\hat{F}_{x^*}, y) \leq \epsilon})$$

for some choice[2] of S and ϵ.

They both *threshold* a score and are algorithmically comparable.

[2] originally $S(\hat{F}_x, y) = d(\eta(y), \eta(y'))$ where $y' \sim F_x$, or MMD etc

History matching and ABC

These methods (anecdotally) seem to work better in mis-specified situations.

Why?

History matching and ABC

These methods (anecdotally) seem to work better in mis-specified situations.

Why?

They differ from likelihood based approaches in that

- They only use some aspect of the simulator output (cf Gibbs posteriors Bissiri *et al.* 2016)
  - Typically we hand pick which simulator outputs to compare, and weight them on a case by case basis.
- Potentially use generalised scores/loss-functions that are more robust to misspecification
- The thresholding type nature potentially makes them somewhat conservative

History matching and ABC

These methods (anecdotally) seem to work better in mis-specified situations.

Why?

They differ from likelihood based approaches in that

- They only use some aspect of the simulator output (cf Gibbs posteriors Bissiri *et al.* 2016)
  - Typically we hand pick which simulator outputs to compare, and weight them on a case by case basis.
- Potentially use generalised scores/loss-functions that are more robust to misspecification
- The thresholding type nature potentially makes them somewhat conservative

$\exists$ many frequentist approaches, minimum scoring rule (Dawid *et al.* 2015, Waghmare *et al.* 2025), etc....

Computing $\pi(x^*|y)$

Cranmer *et al.* 2020

Choice of approximation method depends on

- Computational budget:
  - Max of N simulations possible $\{x_i, f(x_i)\}_{i=1}^N$
- Knowledge of the simulator
  - Zeroth order $f(x)$, first order $\nabla f(x)$, second order $\nabla^2 f(x)$?
  - Intrusive methods? E.g. multi-fidelity/level methods.
  - Known likelihood? Simulation-based inference
- Time-urgency...

Computing $\pi(x^*|y)$

Cranmer *et al.* 2020

Choice of approximation method depends on

- Computational budget:
  - Max of N simulations possible $\{x_i, f(x_i)\}_{i=1}^N$
- Knowledge of the simulator
  - Zeroth order $f(x)$, first order $\nabla f(x)$, second order $\nabla^2 f(x)$?
  - Intrusive methods? E.g. multi-fidelity/level methods.
  - Known likelihood? Simulation-based inference
- Time-urgency...

Distinguish between

- Case-based inference: for each new dataset y, run a separate optimization to approximate $\pi(x^*|y)$.
- Amortized inference: global upfront training (before data collected) using simulations, so that inference at test time is rapid.

Case-based methods

Gold standard remains sampling methods such as MCMC.

- Sequentially simulate $x_1, x_2, \ldots$ from $\pi(x^*|y)$. At stage n

$$x' \sim q(x_n, x')$$

$$\text{Set } x_{n=1} = \begin{cases} x' \text{ w.p. } \alpha(x_n, x', y, f(x', u), f(x, u)) \\ x_n \text{ otherwise} \end{cases}$$

If available, gradient information $\nabla f(x)$ and $\nabla^2 f(x)$ can accelerate mixing.

Case-based methods

Gold standard remains sampling methods such as MCMC.

- Sequentially simulate $x_1, x_2, \ldots$ from $\pi(x^*|y)$. At stage n

$$x' \sim q(x_n, x')$$

$$\text{Set } x_{n=1} = \begin{cases} x' \text{ w.p. } \alpha(x_n, x', y, f(x', u), f(x, u)) \\ x_n \text{ otherwise} \end{cases}$$

  If available, gradient information $\nabla f(x)$ and $\nabla^2 f(x)$ can accelerate mixing.

However, for most complex models MCMC is computationally infeasible without the use of a surrogate $\tilde{f}$.

Case-based methods

Ensemble Kalman methods (Evensen 1994, Iglesias *et al.* 2013, Schillings *et al.* 2017, Garbuno-Inigo *et al.* 2020)

- Exact for linear Gaussian systems. Approximates a Gaussian approximation in non-linear problems...
- Uses a small set of particles $\{x_t^i\}_{i=1}^N$ to approximate posterior at artificial time t.

$$x_{t+1}^i = x_t^i + K_n(y^i - f(x_t^{i+1}))$$

- Can be seen as a derivative-free gradient flow

Case-based methods

Ensemble Kalman methods (Evensen 1994, Iglesias *et al.* 2013, Schillings *et al.* 2017, Garbuno-Inigo *et al.* 2020)

- Exact for linear Gaussian systems. Approximates a Gaussian approximation in non-linear problems...
- Uses a small set of particles $\{x_t^i\}_{i=1}^N$ to approximate posterior at artificial time t.

$$x_{t+1}^i = x_t^i + K_n(y^i - f(x_t^{i+1}))$$

- Can be seen as a derivative-free gradient flow

Variational inference: optimization based alternative to sampling (e.g. Blei *et al.* 2016).

- Choose a parametric family of distributions to approximate posterior, e.g. $q_\phi(x^*) = N(\mu, \Sigma^2)$
- Solve

$$\arg\min_\phi D_{KL}(q_\phi(x^*)||\pi(x^*||y))$$

Surrogate modelling

E.g. Gramacy 2020

If f is slow/costly to evaluate, standard methods such as MCMC are impracticable.

- Inference must be based on ensemble $\mathcal{E} = \{x_n, f(x_n)\}_{n=1}^{N}$.

Surrogate modelling

E.g. Gramacy 2020

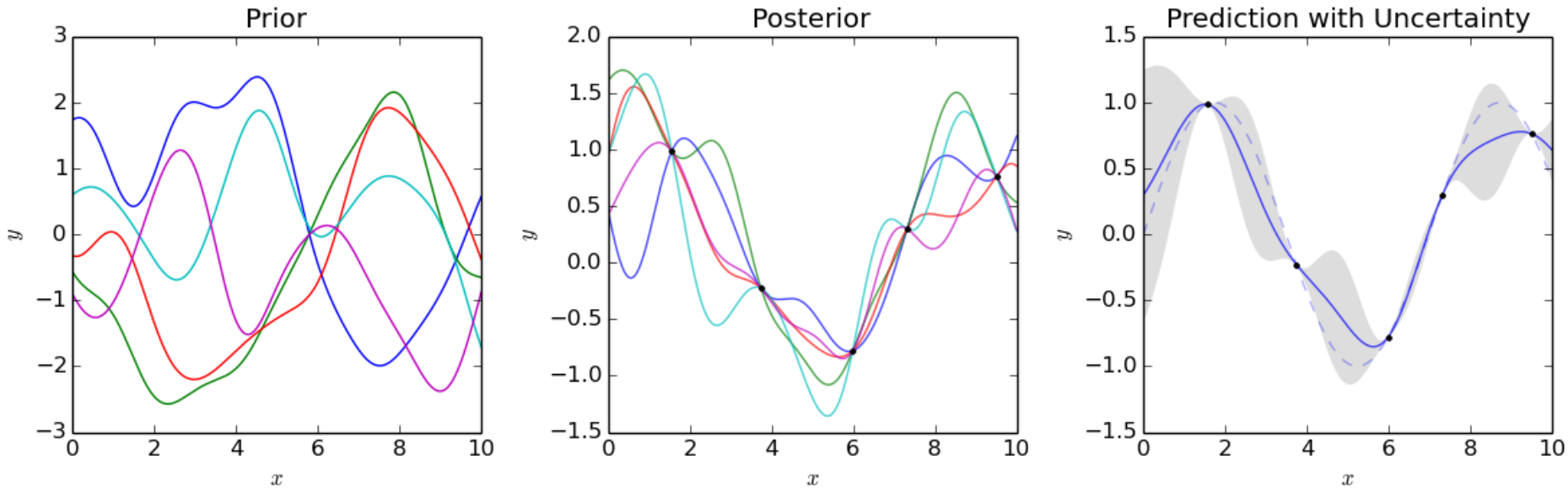
If f is slow/costly to evaluate, standard methods such as MCMC are impracticable.

- Inference must be based on ensemble $\mathcal{E} = \{x_n, f(x_n)\}_{n=1}^{N}$.

Use a surrogate model / emulator / approximation of f, e.g.,

$$f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot))$$

Inference then based on $f(\cdot)|\mathcal{E}$



Note that this adds an additional uncertainty $\pi(f|\mathcal{E})$

Intrusive methods

Methods so far only query $f(x, u)$. But if we can get into the internals there may be model specific structures that can be exploited.

Intrusive methods

Methods so far only query $f(x, u)$. But if we can get into the internals there may be model specific structures that can be exploited.
For example,

- Multi-fidelity or multi-level approaches: if we have $f_1, f_2, \ldots$ that are lower fidelity, but cheaper, versions of f, we can exploit this to reduce costs (eg Peherstorfer *et al.* 2018).

Intrusive methods

Methods so far only query $f(x, u)$. But if we can get into the internals there may be model specific structures that can be exploited.
For example,

- Multi-fidelity or multi-level approaches: if we have $f_1, f_2, \ldots$ that are lower fidelity, but cheaper, versions of f, we can exploit this to reduce costs (eg Peherstorfer *et al.* 2018).
- Linear systems often admit (partially) closed form solution. E.g.

$$\mathcal{L}g = x$$

We can compute $\pi(x|y)$ analytically in some cases (eg Gahungu, Lanyon *et al.* 2022)

Amortized inference

Papamakarios *et al.* 2019, Radev *et al.* 2020

Train a model that predicts $\pi(x^*|y)$ for any y:

- Large upfront cost, rapid test time inference.

Amortized inference

Papamakarios *et al.* 2019, Radev *et al.* 2020

Train a model that predicts $\pi(x^*|y)$ for any y:

- Large upfront cost, rapid test time inference.

Neural posteriors. Eg use a normalizing flow:

- Find invertible g such that

$$x^* \sim \pi(x^*|y) \iff g(x^*; y) \sim N(0, I)$$

then $g^{-1}(z; y) \sim \pi(x^*|y)$ when $z \sim N(0, I)$.

Amortized inference

Papamakarios *et al.* 2019, Radev *et al.* 2020

Train a model that predicts $\pi(x^*|y)$ for any y:

- Large upfront cost, rapid test time inference.

Neural posteriors. Eg use a normalizing flow:

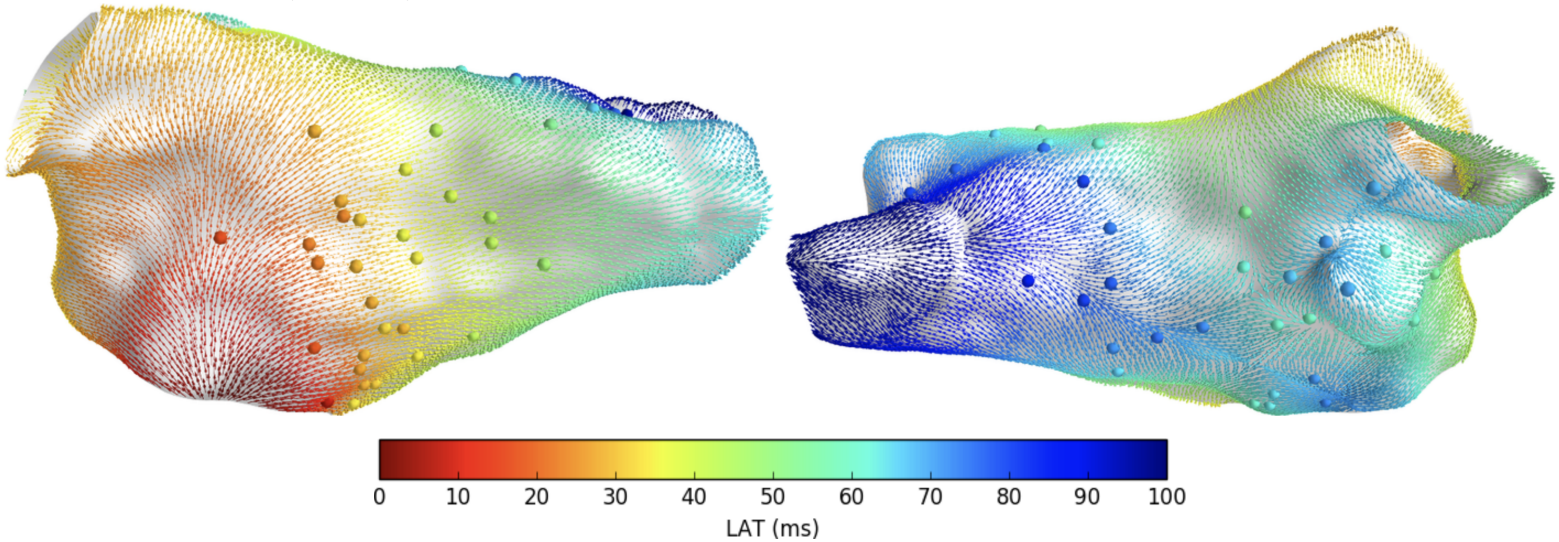
- Find invertible g such that

$$x^* \sim \pi(x^*|y) \iff g(x^*; y) \sim N(0, I)$$

  then $g^{-1}(z; y) \sim \pi(x^*|y)$ when $z \sim N(0, I)$.
- Model g as an invertible NN with easily computable Jacobian.
- Can include an additional summary network $S : y \mapsto \mathbb{R}^p$ to learn optimal summary $\pi(x^*|S(y))$

Motivating example revisited

Coveney *et al.* 2020, 2021, cf Borovitskiy talk in workshop 1

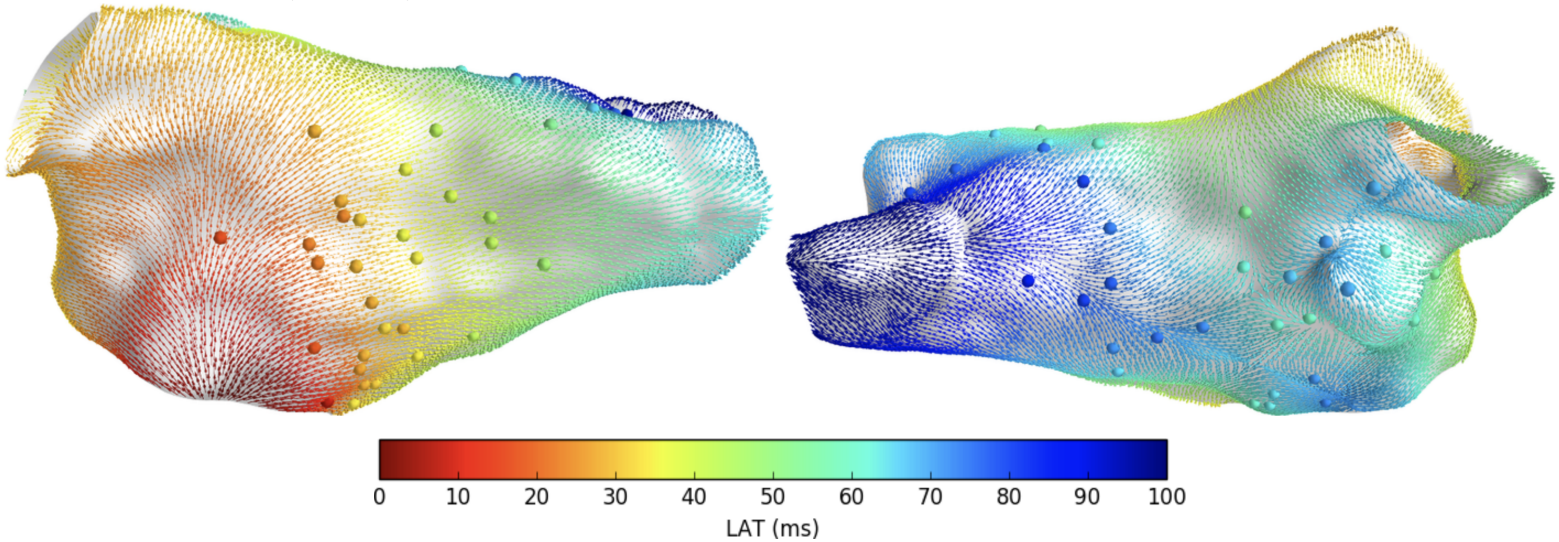


$$LAT(x, w) \sim GP(m(x, w), k((x, w), (x', w')))$$

but standard kernels won't work when domain $w \in \mathcal{G}$ is an atrial manifold

Motivating example revisited

Coveney *et al.* 2020, 2021, cf Borovitskiy talk in workshop 1



$$LAT(x, w) \sim GP(m(x, w), k((x, w), (x', w')))$$

but standard kernels won't work when domain $w \in \mathcal{G}$ is an atrial manifold

Solin *et al.* (2019) showed that if we use a Laplacian eigenbasis

$$-\nabla^2 \phi_j(w) = \lambda_j \phi_j(w) \qquad w \in \mathcal{G}$$

$$\text{then} \quad f(w) = \sum z_k \phi_k(w) \qquad \text{with } z_k \sim N(0, S(\sqrt{\lambda_j}))$$

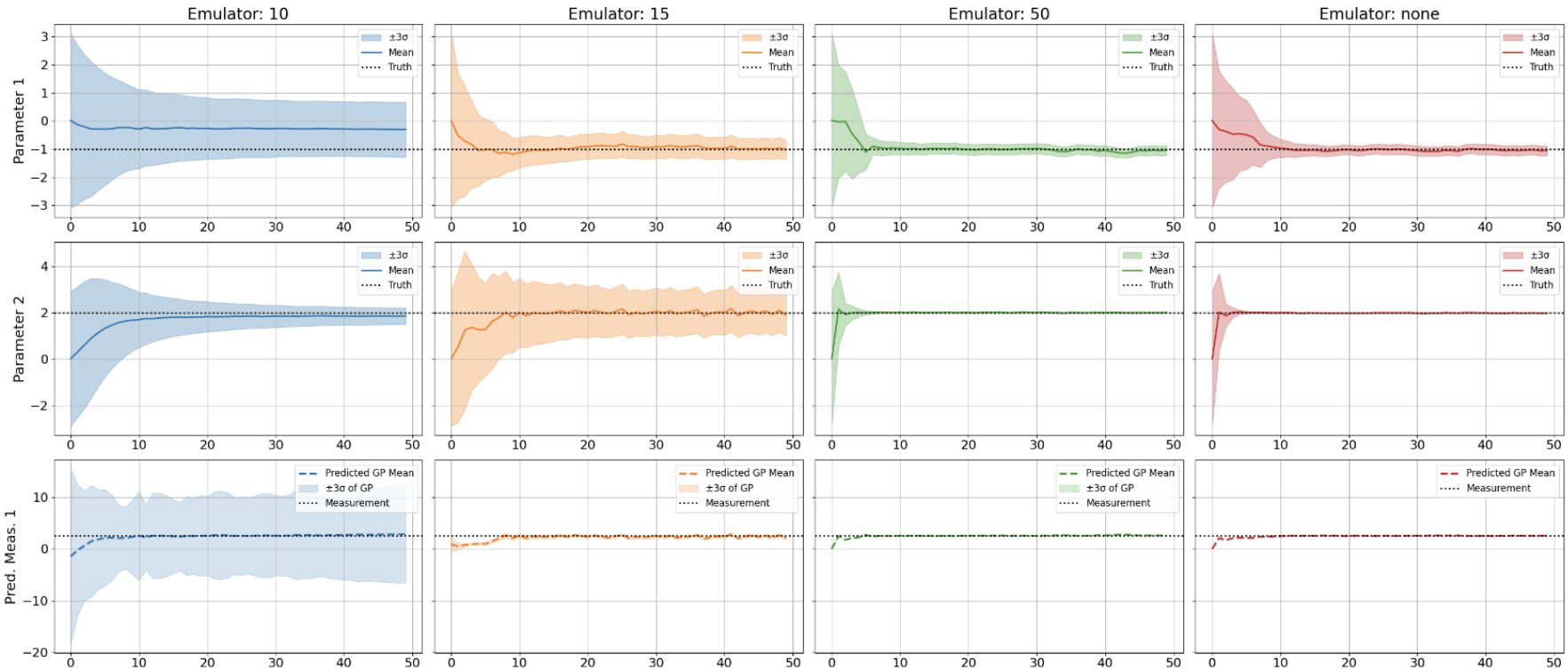
is a GP with spectral density S.

Motivating example revisited

Mamajiwala *et al.* 2024

Can use EnKF with a GP emulator to approximate $\pi(x^*|y)$

- Homogenous parameters
- Works in close to real time
- Can identify 2 (homogeneous) params, but sufficient for prediction of S1S2
- Limited improvement in AF prediction.



Challenges

In digital twin settings, we often have populations of related simulators:

$$f_1, f_2, \ldots \sim \pi(f)$$

For example, $f_i \equiv f_{\mathcal{G}_i}(x, u)$.

Challenges

In digital twin settings, we often have populations of related simulators:

$$f_1, f_2, \ldots \sim \pi(f)$$

For example, $f_i \equiv f_{\mathcal{G}_i}(x, u)$.
How can we reuse simulations from $f_1, \ldots, f_n$ to build a surrogate for f_{n+1}?

- Learn a low dim shape model $w_i = w(\mathcal{G})$ and assume $f_i = f(x, u, w_i)$
- Or learn a general low dim latent representation $f_i = f(x, u, z_i)$
- Learn a discrepancy between instances and a reference model $f_i = f^r(x, u) + \delta_i(x, u)$ where δ_i is simple
- $\vdots$

Challenges

In digital twin settings, we often have populations of related simulators:

$$f_1, f_2, \ldots \sim \pi(f)$$

For example, $f_i \equiv f_{\mathcal{G}_i}(x, u)$.
How can we reuse simulations from $f_1, \ldots, f_n$ to build a surrogate for f_{n+1}?

- Learn a low dim shape model $w_i = w(\mathcal{G})$ and assume $f_i = f(x, u, w_i)$
- Or learn a general low dim latent representation $f_i = f(x, u, z_i)$
- Learn a discrepancy between instances and a reference model $f_i = f^r(x, u) + \delta_i(x, u)$ where δ_i is simple
- $\vdots$

Can we train a single amortized calibration methods for the population?

Conclusions

- High dimensional parameters and expensive simulations means we need to compromise
- Estimating unknown physical values is challenging
- How can we find regularities in the problem to allow us to reduce dimension sufficiently in order to make inference possible?
- How do we use more robust scores to calibrate models with interpretable UQ?
- Can we make amortized methods that work in practical examples?

Conclusions

- High dimensional parameters and expensive simulations means we need to compromise
- Estimating unknown physical values is challenging
- How can we find regularities in the problem to allow us to reduce dimension sufficiently in order to make inference possible?
- How do we use more robust scores to calibrate models with interpretable UQ?
- Can we make amortized methods that work in practical examples?

Thank you for listening!