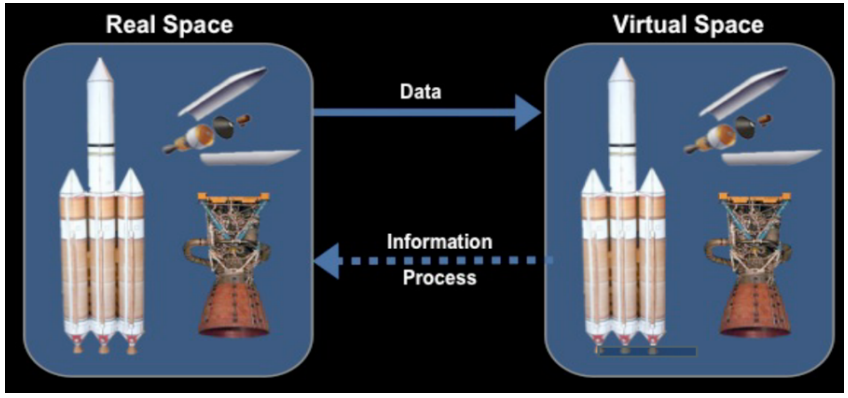
Using GPs to create cardiac digital twins

Richard Wilkinson

University of Nottingham

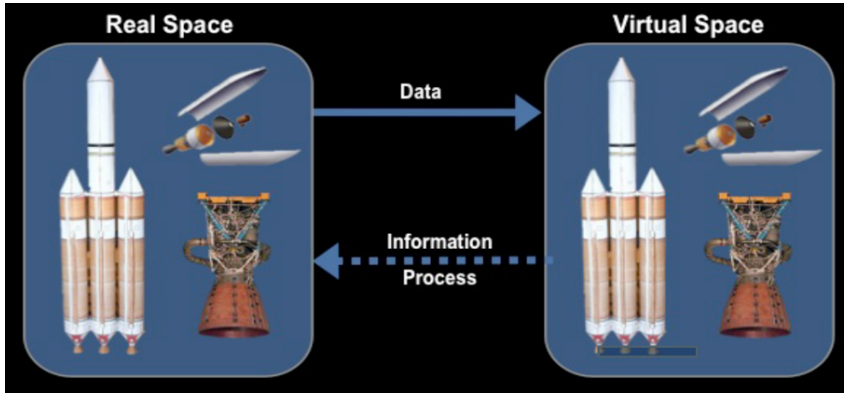
Digital twins

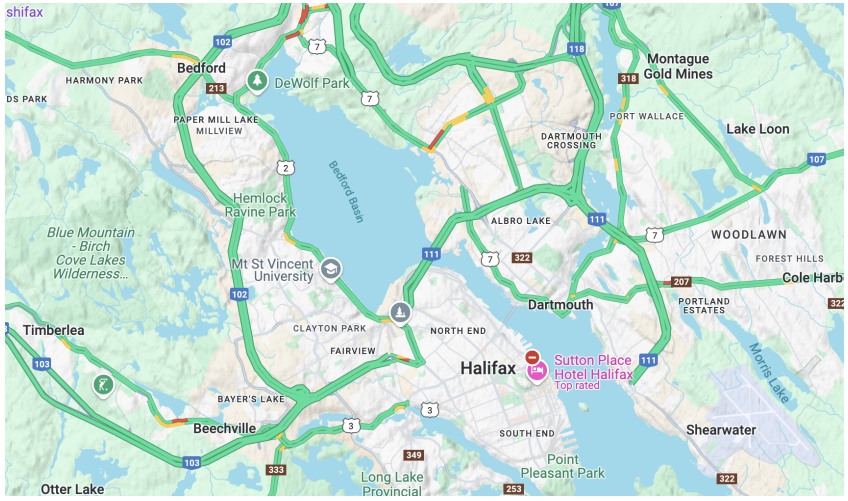
~~*A set of virtual information constructs that mimics the structure, context and behaviour of an individual or unique physical asset, that is dynamically updated with data from its physical twin throughout its life-cycle that informs decisions that realise value.*~~



Digital twins

~~*A set of virtual information constructs that mimics the structure, context and behaviour of an individual or unique physical asset, that is dynamically updated with data from its physical twin throughout its life-cycle that informs decisions that realise value.*~~





A model of an individual, informed by data, that influences decisions.

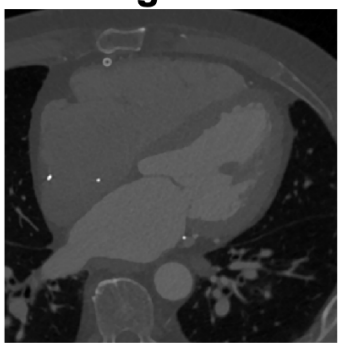
Cardiac physiology

With Steve Niederer, Richard Clayton, Sam Coveney, Cesare Corrardo, Chris Lanyon, Fay Frost, Mariya Mamiwajala, Marina Strocchi, ...

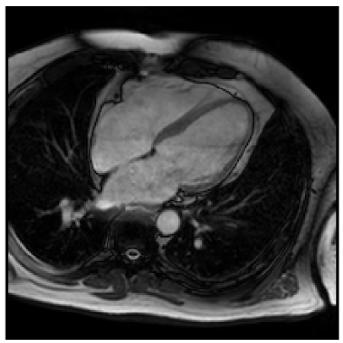
Aim: move from treatment based on guidelines derived from heterogeneous patient groups, to treatment tailored to individual patients based on their data.

Imaging

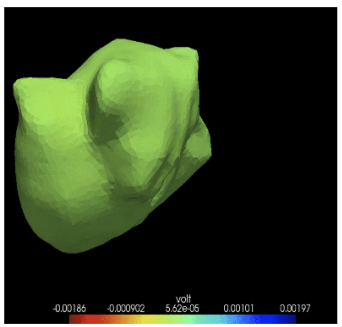
ECG-gated CT



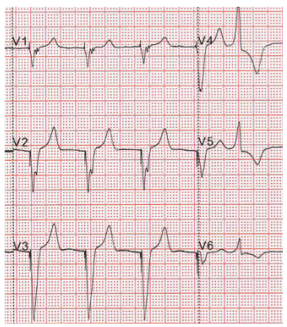
Cardiac MRI



Atrial voltage

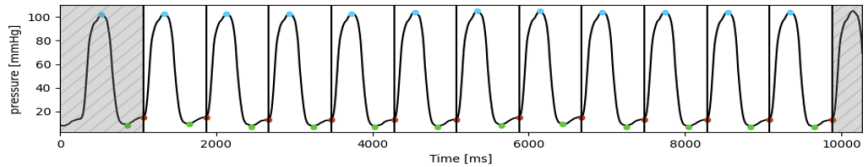


ECGs

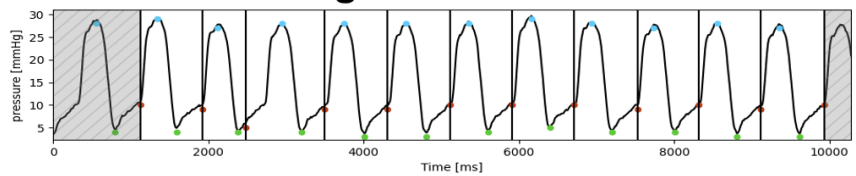


Pressure measurements

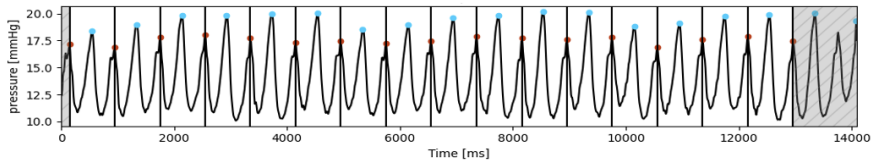
Left ventricle



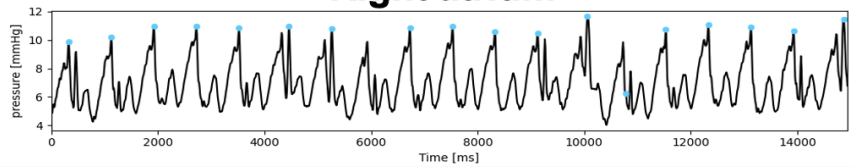
Right ventricle



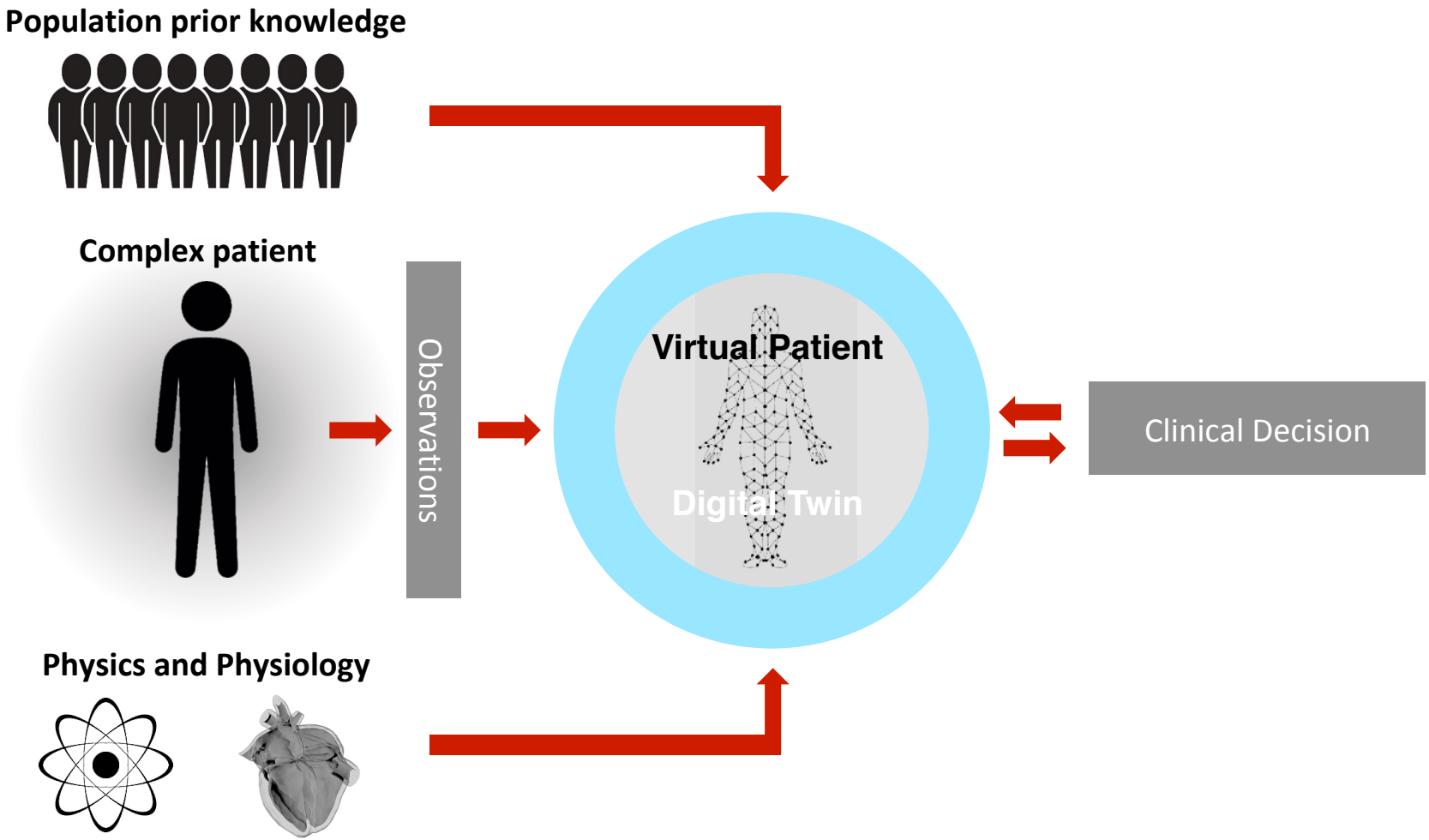
Left atrium



Right atrium



Cardiac digital twin

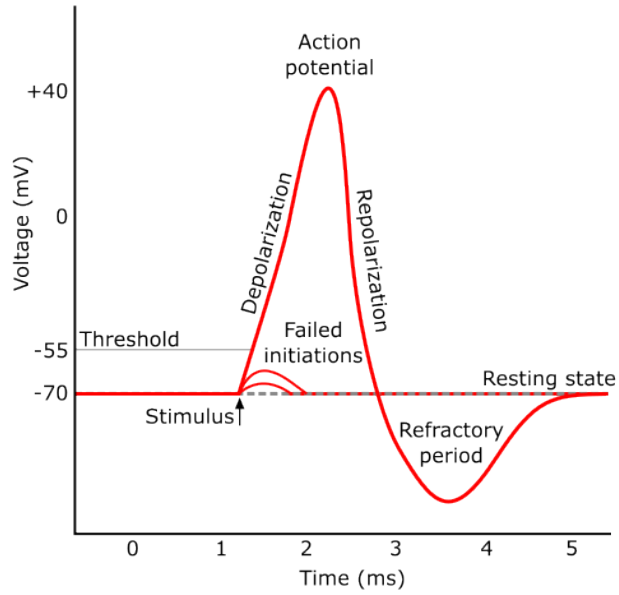


But how confident are we in our prediction

Digital Twins for AF

Digital Twins for AF

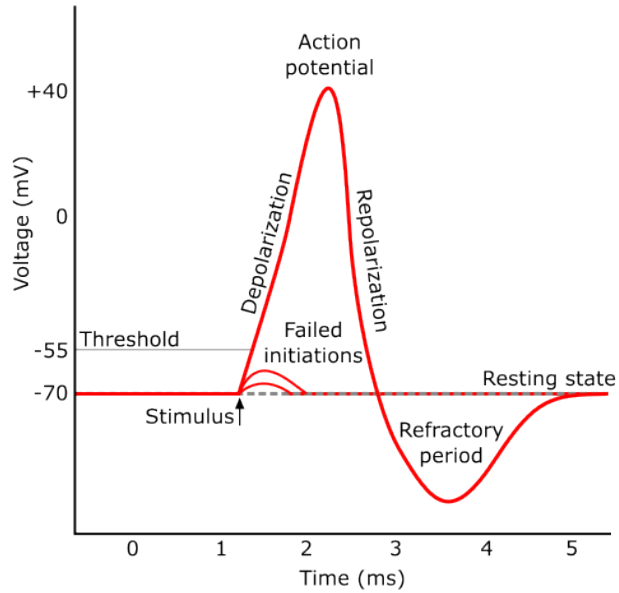
The heart is an electrical-mechanical pump, which contracts under electrical potential.



Digital Twins for AF

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

- Left atrium - sinus rhythm



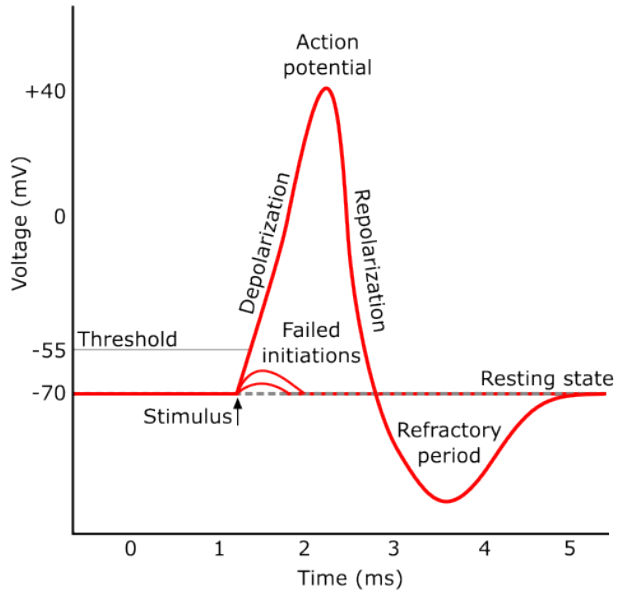
Digital Twins for AF

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.



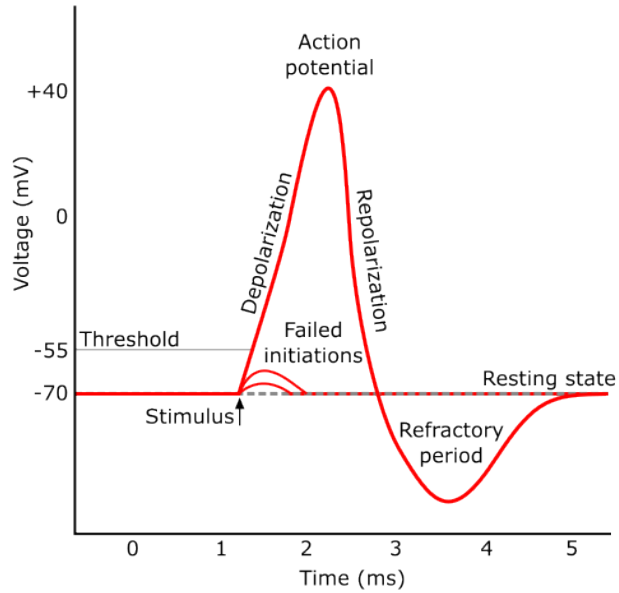
Digital Twins for AF

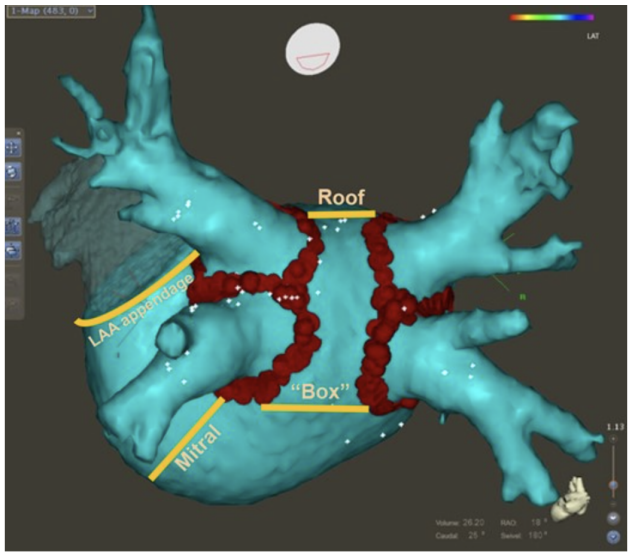
The heart is an electrical-mechanical pump, which contracts under electrical potential.

- 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 $z \in \mathcal{G}$.

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

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

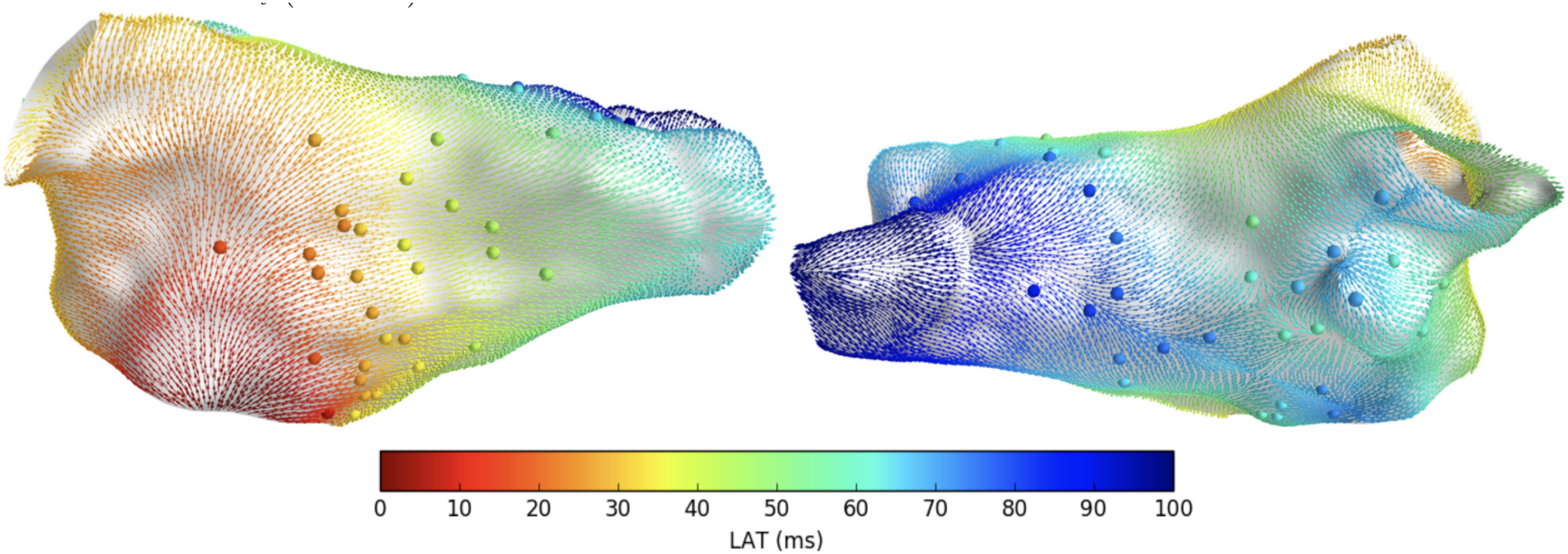
- Parameters $x = \{\tau_{open}(z), \tau_{out}(z), \tau_{in}(z), D(z)\}$
- Control inputs $u_{stim}(z,t)$

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

Predicting AF

Coveney et al. 2022

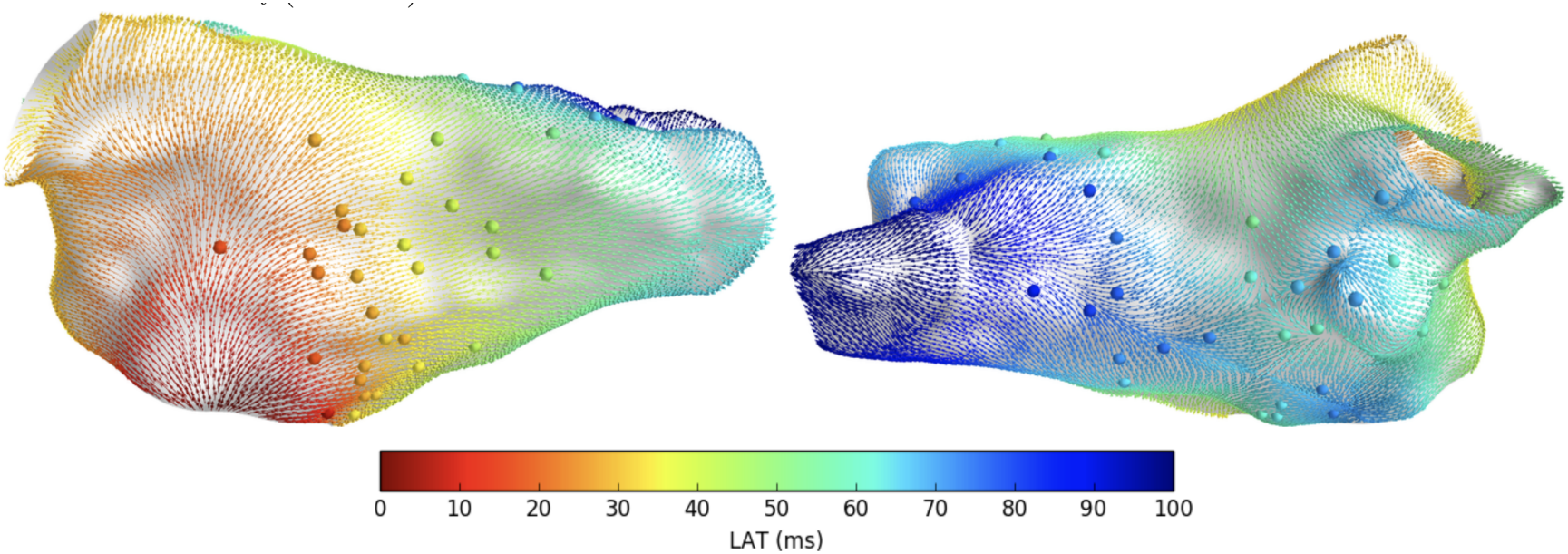
In the clinic, cardiologists pace the heart (i.e. fix u_{stim}) and collect noisy measurements of local activation times (LAT) at some locations.



Predicting AF

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

Challenges of calibrating cardiac digital twins

Complex inference problem

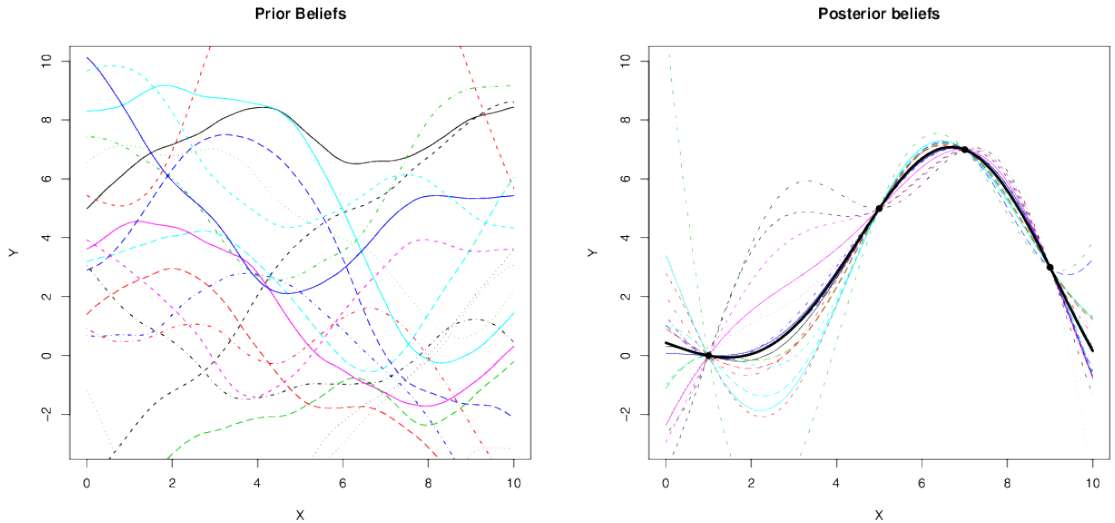
- High dimensional parameter x with sparse noisy data y
- Expensive simulator f
- Uncertain geometry

To be a practical clinical tool inference needs to be fast, cheap, and scalable

GPs can help!

Quick recap: Gaussian processes (GP)

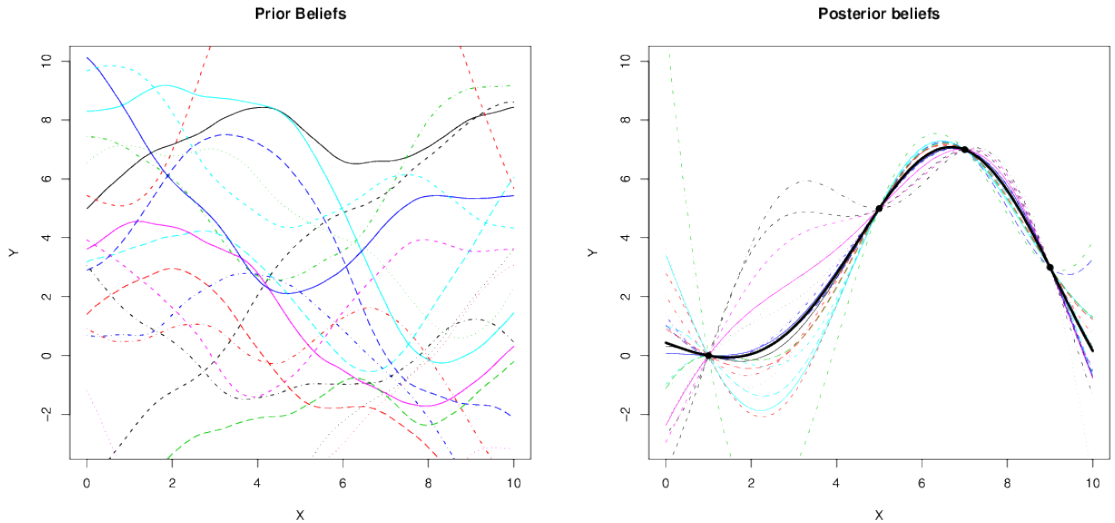
Regression: given data $\{x_i, y_i = f(x_i)\}_{i=1}^n$ learn f.



GPs can be thought of as probabilistic models of functions.

Quick recap: Gaussian processes (GP)

Regression: given data $\{x_i, y_i = f(x_i)\}_{i=1}^n$ learn f.



GPs can be thought of as probabilistic models of functions.

- f, a random process indexed by $x \in \mathcal{X}$, such that for $x_1, \ldots, x_n$,

$$\mathbf{f} = (f(x_1), \ldots, f(x_n)) \sim N_n(\mathbf{m}, \mathbf{K})$$

where $K_{ij} = k(x_i, x_j)$

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

Key choice is the covariance/kernel function $k(x, x') = \mathbb{C}\text{ov}(f(x), f(x'))$

Why use GPs? Answer 1

The GP class of models is closed under various operations.

Why use GPs? Answer 1

The GP class of models is closed under various operations.

- Closed under Bayesian conditioning, i.e., if we observe

$$\mathbf{D} = (f(x_1), \ldots, f(x_n))$$

then

$$f|D \sim GP$$

but with updated mean and covariance functions.

Why use GPs? Answer 1

The GP class of models is closed under various operations.

- Closed under Bayesian conditioning, i.e., if we observe

$$\mathbf{D} = (f(x_1), \ldots, f(x_n))$$

then

$$f|D \sim GP$$

but with updated mean and covariance functions.

- Closed under addition

$$f_1(\cdot), f_2(\cdot) \sim GP \quad \text{then} \quad (f_1 + f_2)(\cdot) \sim GP$$

Why use GPs? Answer 1

The GP class of models is closed under various operations.

- Closed under Bayesian conditioning, i.e., if we observe

$$\mathbf{D} = (f(x_1), \ldots, f(x_n))$$

then

$$f|D \sim GP$$

but with updated mean and covariance functions.

- Closed under addition

$$f_1(\cdot), f_2(\cdot) \sim GP \quad \text{then} \quad (f_1 + f_2)(\cdot) \sim GP$$

- Closed under any linear operator. If $f \sim GP(m(\cdot), k(\cdot,\cdot))$, then if $\mathcal{L}$ is a linear operator

$$\mathcal{L} \circ f \sim GP(\mathcal{L} \circ m, \mathcal{L}^2 \circ k)$$

e.g. $\frac{df}{dx}$, $\int f(x)dx$, Af are all GPs

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \sigma^2 ||\beta||_2^2 \quad \text{regularised least squares}$$

$$\text{where} \quad X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix}$$

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \sigma^2 ||\beta||_2^2 \quad \text{regularised least squares}$$
$$= (X^\top X + \sigma^2 I)^{-1} X^\top y \quad \text{usual ridge regression estimator}$$

$$\text{where} \quad X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix}$$

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

$$\begin{aligned}
\hat{\beta} &= \arg\min_{\beta} ||y - X\beta||_2^2 + \sigma^2 ||\beta||_2^2 \quad \text{regularised least squares} \\
&= (X^\top X + \sigma^2 I)^{-1} X^\top y \quad \text{usual ridge regression estimator} \\
&= X^\top (XX^\top + \sigma^2 I)^{-1} y \quad \text{the dual form}
\end{aligned}$$

$$\text{where} \quad X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix}$$

Why use GPs? Answer 2: non-parametric/kernel regression

We can also view GPs as a non-parametric extension to linear regression.

- k determines the space of functions that sample paths live in.

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \sigma^2 ||\beta||_2^2 \quad \text{regularised least squares}$$
$$= (X^\top X + \sigma^2 I)^{-1} X^\top y \quad \text{usual ridge regression estimator}$$
$$= X^\top (XX^\top + \sigma^2 I)^{-1} y \quad \text{the dual form}$$

as $$(X^\top X + \sigma^2 I) X^\top = X^\top (XX^\top + \sigma^2 I)$$

so $$X^\top (XX^\top + \sigma^2 I)^{-1} = (X^\top X + \sigma^2 I)^{-1} X^\top$$

where $$X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix}$$

At first the dual form

$$\hat{\beta} = X^\top (XX^\top + \sigma^2 I)^{-1} y$$

looks harder to compute than the usual

$$\hat{\beta} = (X^\top X + \sigma^2 I)^{-1} X^\top y$$

- $X^\top X$ is $p \times p$ $\quad$ $p =$ number of features/parameters
- $XX^\top$ is $n \times n$ $\quad$ n is the number of data points

At first the dual form

$$\hat{\beta} = X^\top (XX^\top + \sigma^2 I)^{-1} y$$

looks harder to compute than the usual

$$\hat{\beta} = (X^\top X + \sigma^2 I)^{-1} X^\top y$$

- $X^\top X$ is $p \times p$ $\quad$ $p =$ number of features/parameters
- $XX^\top$ is $n \times n$ $\quad$ n is the number of data points

But the dual form only uses inner products between vectors in $\mathbb{R}^n$

$$XX^\top = \begin{pmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{pmatrix} (x_1 \dots x_n) = \begin{pmatrix} x_1^\top x_1 & \dots & x_1^\top x_n \\ \vdots & & \\ x_n^\top x_1 & \dots & x_n^\top x_n \end{pmatrix}$$

$$= K_{XX} \text{ if } k(x, x') = x^\top x'$$

— This is useful!

Prediction

The best prediction of y at a new location x' is

$$\begin{aligned}\hat{y}' &= x'^{\top}\hat{\beta} \\ &= x'^{\top}X^{\top}(XX^{\top}+\sigma^2 I)^{-1}y \\ &= k_X(x')^{\top}(K_{XX}+\sigma^2 I)^{-1}y\end{aligned}$$

where $k_X(x')^{\top} := (x'^{\top}x_1, \ldots, x'^{\top}x_n)$ and $[K_{XX}]_{ij} := x_i^{\top}x_j$

Prediction

The best prediction of y at a new location x' is

$$\begin{aligned}\hat{y}' &= x'^\top \hat{\beta} \\ &= x'^\top X^\top (XX^\top + \sigma^2 I)^{-1} y \\ &= k_X(x')^\top (K_{XX} + \sigma^2 I)^{-1} y\end{aligned}$$

where $k_X(x')^\top := (x'^\top x_1, \ldots, x'^\top x_n)$ and $[K_{XX}]_{ij} := x_i^\top x_j$
K_{XX} and $k_X(x)$ are kernel matrices:

- every element is an inner product between 2 points: $k(x, x') = x^\top x'$

Prediction

The best prediction of y at a new location x' is

$$\begin{aligned}\hat{y}' &= x'^\top \hat{\beta} \\ &= x'^\top X^\top (XX^\top + \sigma^2 I)^{-1} y \\ &= k_X(x')^\top (K_{XX} + \sigma^2 I)^{-1} y\end{aligned}$$

where $k_X(x')^\top := (x'^\top x_1, \ldots, x'^\top x_n)$ and $[K_{XX}]_{ij} := x_i^\top x_j$
K_{XX} and $k_X(x)$ are kernel matrices:

- every element is an inner product between 2 points: $k(x, x') = x^\top x'$

Note this is the GP conditional mean when $m(x) = 0$.

$$m(x) = k_X(x)^\top (K_{XX} + \sigma^2 I)^{-1} y$$

- linear regression and GP regression are equivalent when $k(x, x') = x^\top x'$.

Including features I

We can replace x by a feature vector in linear regression, e.g., $\phi(x) = (1\ x\ x^2)$

Only the inner product changes:

$$k(x', x) = x'^\top x$$

is replaced by

$$k(x', x) = \phi(x')^\top \phi(x)$$

Including features I

We can replace x by a feature vector in linear regression, e.g., $\phi(x) = (1\ x\ x^2)$

Only the inner product changes:

$$k(x', x) = x'^{\top} x$$

is replaced by

$$k(x', x) = \phi(x')^{\top} \phi(x)$$

Note $k(x', x) = \phi(x')^{\top} \phi(x)$ is a positive semi-definite function for any choice of $\phi(x)$.

Including features II

For some sets of features, $\phi(x)$, computation of the inner product doesn't require us to evaluate the individual features.

Including features II

For some sets of features, $\phi(x)$, computation of the inner product doesn't require us to evaluate the individual features.

E.g., Consider $\mathcal{X} = \mathbb{R}^2$ and let

$$\phi : \mathbf{x} = (x_1, x_2) \mapsto (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$$

i.e., linear regression using all the linear and quadratic terms, and first order interactions.

Including features II

For some sets of features, $\phi(x)$, computation of the inner product doesn't require us to evaluate the individual features.

E.g., Consider $\mathcal{X} = \mathbb{R}^2$ and let

$$\phi : \mathbf{x} = (x_1, x_2) \mapsto (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$$

i.e., linear regression using all the linear and quadratic terms, and first order interactions.
Then

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= \phi(x)^\top \phi(z) \\ &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1z_2, z_2^2)^\top \\ &= (1 + (x_1, x_2)(z_1, z_2)^\top)^2 \\ &= (1 + \mathbf{x}^\top \mathbf{z})^2 \end{aligned}$$

Including features II

For some sets of features, $\phi(x)$, computation of the inner product doesn't require us to evaluate the individual features.

E.g., Consider $\mathcal{X} = \mathbb{R}^2$ and let

$$\phi : \mathbf{x} = (x_1, x_2) \mapsto (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$$

i.e., linear regression using all the linear and quadratic terms, and first order interactions.
Then

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= \phi(x)^\top \phi(z) \\ &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1z_2, z_2^2)^\top \\ &= (1 + (x_1, x_2)(z_1, z_2)^\top)^2 \\ &= (1 + \mathbf{x}^\top \mathbf{z})^2 \end{aligned}$$

To evaluate $k(\mathbf{x}, \mathbf{z})$ we didn't need to explicitly compute the feature vector $\phi(\mathbf{x})$

Including features III

To evaluate $k(\mathbf{x}, \mathbf{z})$ we didn't need to explicitly compute the feature vectors $\phi(\mathbf{z}) \in \mathbb{R}^6$

The same idea works with much larger feature vectors, sometimes even when $\phi(\mathbf{x}) \in \mathbb{R}^\infty$

Including features III

To evaluate $k(\mathbf{x}, \mathbf{z})$ we didn't need to explicitly compute the feature vectors $\phi(\mathbf{z}) \in \mathbb{R}^6$
The same idea works with much larger feature vectors, sometimes even when $\phi(\mathbf{x}) \in \mathbb{R}^\infty$
Theorem: A function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

is positive semi-definite (and thus a valid covariance function) if and only if we can write

$$k(x, x') = \phi(x)^\top \phi(x')$$

for some (possibly infinite dimensional) feature vector $\phi(x)$.

Including features III

To evaluate $k(\mathbf{x}, \mathbf{z})$ we didn't need to explicitly compute the feature vectors $\phi(\mathbf{z}) \in \mathbb{R}^6$
The same idea works with much larger feature vectors, sometimes even when $\phi(\mathbf{x}) \in \mathbb{R}^\infty$
Theorem: A function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

is positive semi-definite (and thus a valid covariance function) if and only if we can write

$$k(x, x') = \phi(x)^\top \phi(x')$$

for some (possibly infinite dimensional) feature vector $\phi(x)$.

So GP regression with k can be thought of as linear regression with an infinite $\phi(x)$

Kernel regression (see Kanagawa et al. 2019)

Kernel regression and GP regression are closely related.

Kernel regression (see Kanagawa et al. 2019)

Kernel regression and GP regression are closely related.
Consider the space of functions

$$\mathcal{H}_k = \overline{\text{span}}\{k(\cdot, x) : x \in \mathcal{X}\}$$

ie functions of the form $\sum_{i=1}^{n} \alpha_i k(x, x_i)$ with inner product

$$\langle \sum a_i k(\cdot, x_i), \sum b_i k(\cdot, y_i) \rangle = \sum_{ij} a_i b_j k(x_i, y_j)$$

Kernel regression (see Kanagawa et al. 2019)

Kernel regression and GP regression are closely related.
Consider the space of functions

$$\mathcal{H}_k = \overline{\text{span}}\{k(\cdot, x) : x \in \mathcal{X}\}$$

ie functions of the form $\sum_{i=1}^{n} \alpha_i k(x, x_i)$ with inner product

$$\langle \sum a_i k(\cdot, x_i), \sum b_i k(\cdot, y_i) \rangle = \sum_{ij} a_i b_j k(x_i, y_j)$$

This is the reproducing kernel Hilbert space (RKHS) associated with k.

Kernel regression (see Kanagawa et al. 2019)

Kernel regression and GP regression are closely related.
Consider the space of functions

$$\mathcal{H}_k = \overline{\text{span}}\{k(\cdot, x) : x \in \mathcal{X}\}$$

ie functions of the form $\sum_{i=1}^{n} \alpha_i k(x, x_i)$ with inner product

$$\langle \sum a_i k(\cdot, x_i), \sum b_i k(\cdot, y_i) \rangle = \sum_{ij} a_i b_j k(x_i, y_j)$$

This is the reproducing kernel Hilbert space (RKHS) associated with k.
Kernel ridge regression chooses $f \in \mathcal{H}_k$ to minimise

$$L(f) = \sum_i (f(x_i) - y_i)^2 + \sigma^2 ||f||^2_{\mathcal{H}_k}$$

Kernel regression (see Kanagawa et al. 2019)

Kernel regression and GP regression are closely related.
Consider the space of functions

$$\mathcal{H}_k = \overline{\text{span}}\{k(\cdot, x) : x \in \mathcal{X}\}$$

ie functions of the form $\sum_{i=1}^{n} \alpha_i k(x, x_i)$ with inner product

$$\langle \sum a_i k(\cdot, x_i), \sum b_i k(\cdot, y_i) \rangle = \sum_{ij} a_i b_j k(x_i, y_j)$$

This is the reproducing kernel Hilbert space (RKHS) associated with k.
Kernel ridge regression chooses $f \in \mathcal{H}_k$ to minimise

$$L(f) = \sum_i (f(x_i) - y_i)^2 + \sigma^2 ||f||^2_{\mathcal{H}_k}$$

We can show that

$$\bar{m}(x) = \arg \min_{f \in \mathcal{H}_k} L(f)$$

where $\bar{m}(x)$ is the same as the GP posterior mean

TL;DR

Functions live in function spaces (vector spaces with inner products). There are lots of different function spaces: the GP kernel implicitly determines which particular (RKHS) space we work with - our hypothesis space.

- Generally, we don't think too hard about this space, we just choose a kernel and attempt to validate it empirically.

[1] and can be dense in some sets of continuous bounded functions

TL;DR

Functions live in function spaces (vector spaces with inner products). There are lots of different function spaces: the GP kernel implicitly determines which particular (RKHS) space we work with - our hypothesis space.

- Generally, we don't think too hard about this space, we just choose a kernel and attempt to validate it empirically.

Although reality may not lie in the RKHS defined by k, this space is much richer than any parametric regression model[1],

- thus is more likely to contain an element close to the true functional form than any class of models that contains only a finite number of features.

This is the motivation for non-parametric methods.

[1] and can be dense in some sets of continuous bounded functions

Why use GPs? Answer 3: Naturalness of GP framework

Kriging

Suppose $Y(x)$ is a (second order stationary) stochastic process with

$$\mathbb{E}Y(x) = \mu \; \forall \; x$$
$$\mathbb{C}\text{ov}(Y(x), Y(x')) = k(x - x') \; \forall \; x, x'$$

NB we're not assuming Y has a Gaussian distribution.

Why use GPs? Answer 3: Naturalness of GP framework

Kriging

Suppose $Y(x)$ is a (second order stationary) stochastic process with

$$\mathbb{E}Y(x) = \mu \; \forall \; x$$
$$\mathbb{C}\text{ov}(Y(x), Y(x')) = k(x - x') \; \forall \; x, x'$$

NB we're not assuming Y has a Gaussian distribution.

If someone tells you $\mathbf{y} = (Y(x_1), \ldots, Y(x_n))^\top$, how would you predict $Y(x)$?

Why use GPs? Answer 3: Naturalness of GP framework

Kriging

Suppose $Y(x)$ is a (second order stationary) stochastic process with

$$\mathbb{E}Y(x) = \mu \; \forall \; x$$
$$\mathbb{C}\text{ov}(Y(x), Y(x')) = k(x - x') \; \forall \; x, x'$$

NB we're not assuming Y has a Gaussian distribution.

If someone tells you $\mathbf{y} = (Y(x_1), \ldots, Y(x_n))^\top$, how would you predict $Y(x)$?
One option is to find the best linear unbiased predictor (BLUP) of $Y(x)$.

Best Linear Unbiased Predictors (BLUP)

Consider the linear estimator

$$\hat{Y}(x) = c + \sum w_i Y(x_i) = c + \mathbf{w}^\top \mathbf{y}$$

Best Linear Unbiased Predictors (BLUP)

Consider the linear estimator

$$\hat{Y}(x) = c + \sum w_i Y(x_i) = c + \mathbf{w}^\top \mathbf{y}$$

If we require $\hat{Y}(x)$ to be unbiased,

$$\begin{aligned} \mu &= \mathbb{E}\hat{Y}(x) \\ &= \mathbb{E}(c + \mathbf{w}^\top \mathbf{y}) \\ &= c + \mathbf{w}^\top \boldsymbol{\mu} \end{aligned}$$

where $\boldsymbol{\mu} = (\mu, \ldots, \mu)^\top$.

Best Linear Unbiased Predictors (BLUP)

Consider the linear estimator

$$\hat{Y}(x) = c + \sum w_i Y(x_i) = c + \mathbf{w}^\top \mathbf{y}$$

If we require $\hat{Y}(x)$ to be unbiased,

$$\begin{aligned} \mu &= \mathbb{E}\hat{Y}(x) \\ &= \mathbb{E}(c + \mathbf{w}^\top \mathbf{y}) \\ &= c + \mathbf{w}^\top \boldsymbol{\mu} \end{aligned}$$

where $\boldsymbol{\mu} = (\mu, \ldots, \mu)^\top$.

Thus $c = \mu - \mathbf{w}^\top \boldsymbol{\mu}$ and we must have

$$\hat{Y}(x) = \mu + \mathbf{w}^\top (\mathbf{y} - \boldsymbol{\mu})$$

Best Linear Unbiased Predictors (BLUP) - II

The best linear unbiased predictor minimises the mean square error

$$
\begin{aligned}
MSE(\hat{Y}(x)) &= \mathbb{E}((\hat{Y}(x) - Y(x))^2) \\
&= \mathbb{E}\left((\mathbf{w}^\top(\mathbf{y} - \boldsymbol{\mu}) + (\mu - Y(x))^2\right) \\
&= \mathbf{w}^\top \mathbb{V}\mathrm{ar}(\mathbf{y})\mathbf{w} + \mathbb{V}\mathrm{ar}(Y(x)) - 2\mathbf{w}^\top \mathbb{C}\mathrm{ov}(\mathbf{y}, Y(x)) \\
&= \mathbf{w}^\top K_{XX}\mathbf{w} + k(0) - 2\mathbf{w}^\top \mathbf{k}_X(x)
\end{aligned}
$$

Best Linear Unbiased Predictors (BLUP) - II

The best linear unbiased predictor minimises the mean square error

$$\begin{aligned} MSE(\hat{Y}(x)) &= \mathbb{E}((\hat{Y}(x) - Y(x))^2) \\ &= \mathbb{E}\left((\mathbf{w}^\top(\mathbf{y} - \boldsymbol{\mu}) + (\mu - Y(x))^2\right) \\ &= \mathbf{w}^\top \mathbb{V}\text{ar}(\mathbf{y})\mathbf{w} + \mathbb{V}\text{ar}(Y(x)) - 2\mathbf{w}^\top \mathbb{C}\text{ov}(\mathbf{y}, Y(x)) \\ &= \mathbf{w}^\top K_{XX}\mathbf{w} + k(0) - 2\mathbf{w}^\top \mathbf{k}_X(x) \end{aligned}$$

If we differentiate wrt w and set the gradient equal to zero, we find

$$0 = 2K_{XX}\mathbf{w} - 2\mathbf{k}_X(x)$$

Best Linear Unbiased Predictors (BLUP) - II

The best linear unbiased predictor minimises the mean square error

$$\begin{aligned} MSE(\hat{Y}(x)) &= \mathbb{E}((\hat{Y}(x) - Y(x))^2) \\ &= \mathbb{E}\left((\mathbf{w}^\top(\mathbf{y} - \boldsymbol{\mu}) + (\mu - Y(x))^2\right) \\ &= \mathbf{w}^\top \mathbb{V}\mathrm{ar}(\mathbf{y})\mathbf{w} + \mathbb{V}\mathrm{ar}(Y(x)) - 2\mathbf{w}^\top \mathbb{C}\mathrm{ov}(\mathbf{y}, Y(x)) \\ &= \mathbf{w}^\top K_{XX}\mathbf{w} + k(0) - 2\mathbf{w}^\top \mathbf{k}_X(x) \end{aligned}$$

If we differentiate wrt w and set the gradient equal to zero, we find

$$0 = 2K_{XX}\mathbf{w} - 2\mathbf{k}_X(x)$$

and thus

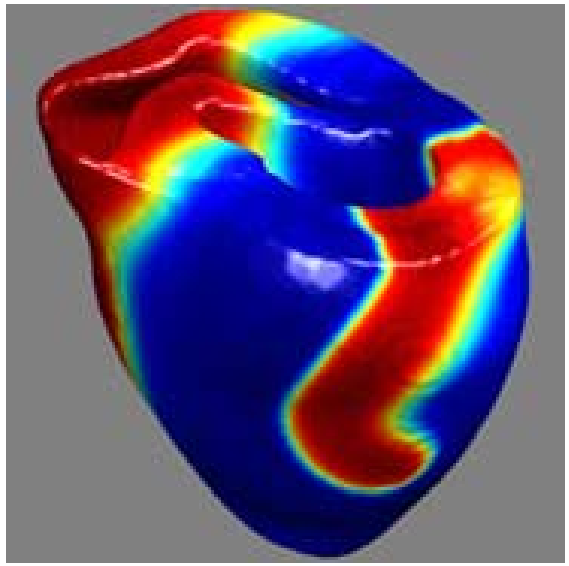
$$\hat{Y}(x) = \mu + \mathbf{k}_X(x)^\top K_{XX}^{-1}(\mathbf{y} - \mu)$$

as before.

So the Gaussian process posterior mean is optimal (i.e. is the BLUP) even if we don't assume Gaussianity.

Problem 1: GPs on manifolds

Coveney *et al.* *IEEE TBME* 2019



We want to estimate

- the time of arrival of the wave front - the Local Activation Time (LAT)
- the wave's Conduction Velocity (CV)

using data from an Electrophysiology (EP) study:

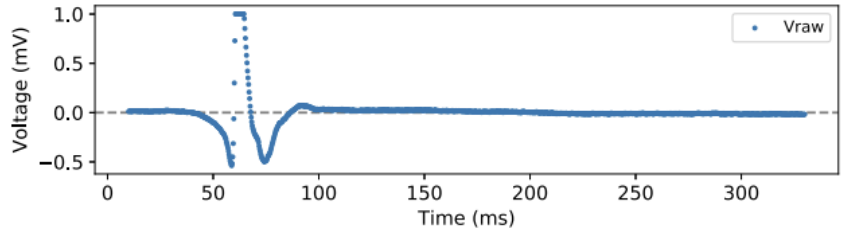
- electrodes placed on the surface of the atrium and electrical pacing applied at various frequencies.

Estimating local activation times from electrograms

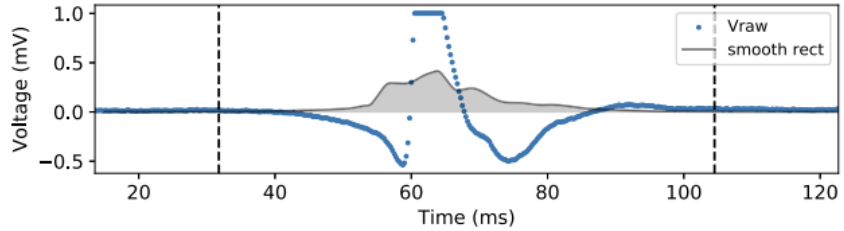
Coveney *et al.* *IEEE TBME* 2019

How should local activation times (LAT) be inferred from a clipped bipolar electrogram?

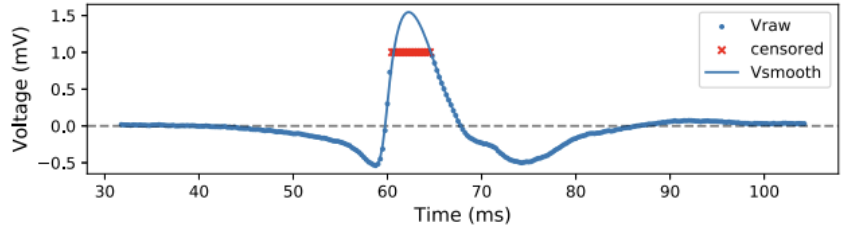
- We model the local ECG as $V(t)$ and infer the position of the maximum, accounting for the clipped (censored) voltage trace.



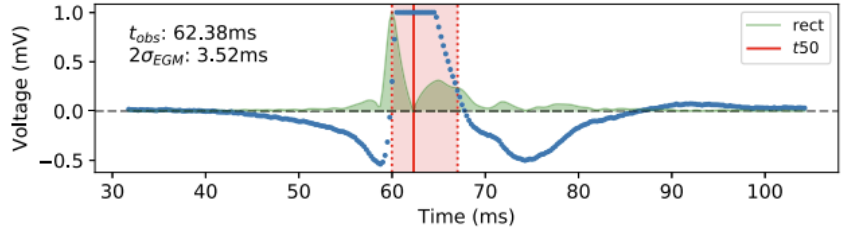
(a) **Regularize signal**. Blue points show $V_{raw}(t)$, the signal minus the mean with normalized amplitude. Recording has clipped the peak.



(b) **Bracket activation complex**. EGM is smoothed, rectified, and smoothed again (black line/area). Vertical dashed lines are brackets.



(c) **Smooth the signal**. Gaussian Process fit to $V_{raw}(t)$, ignoring censored data (red points), giving $V_{smooth}(t)$ (smoothed and *reconstructed*).

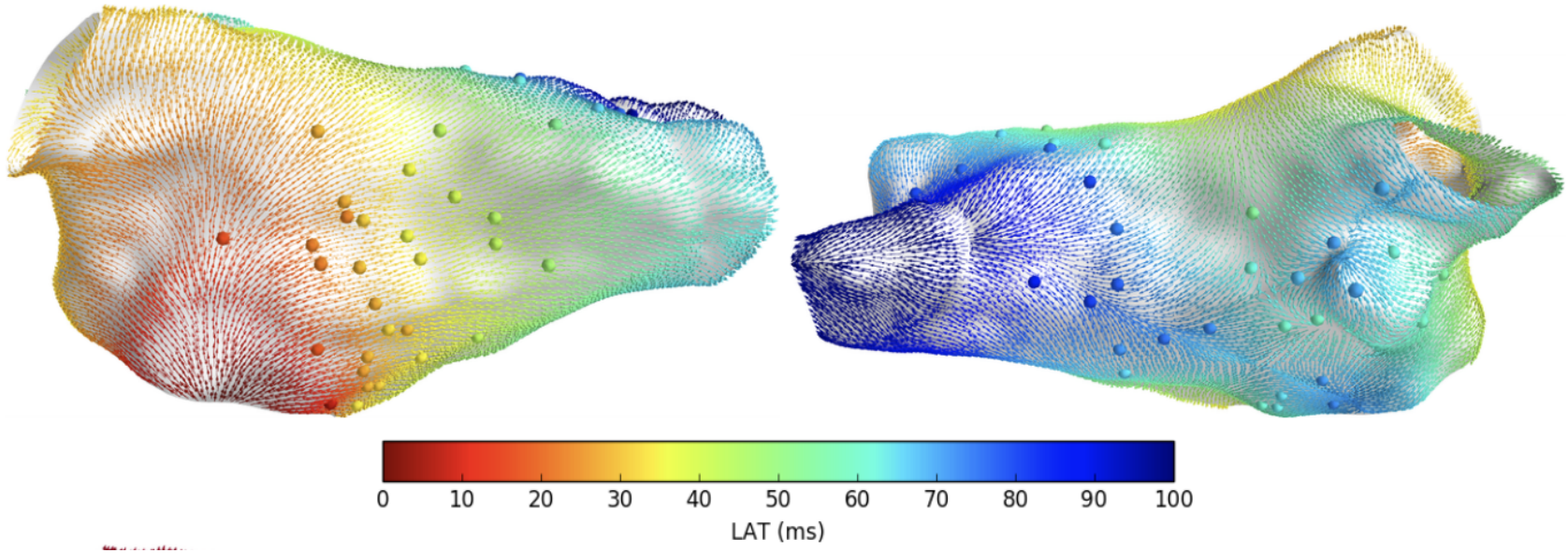


(d) **Assign uncertain LAT**. Rectified smoothed signal (green area). The solid red line is t_{50}, shaded area edges are t_{25} and t_{75}.

Interpolation between locations

We want to estimate activation times at all locations on the atria (the *LAT map*)

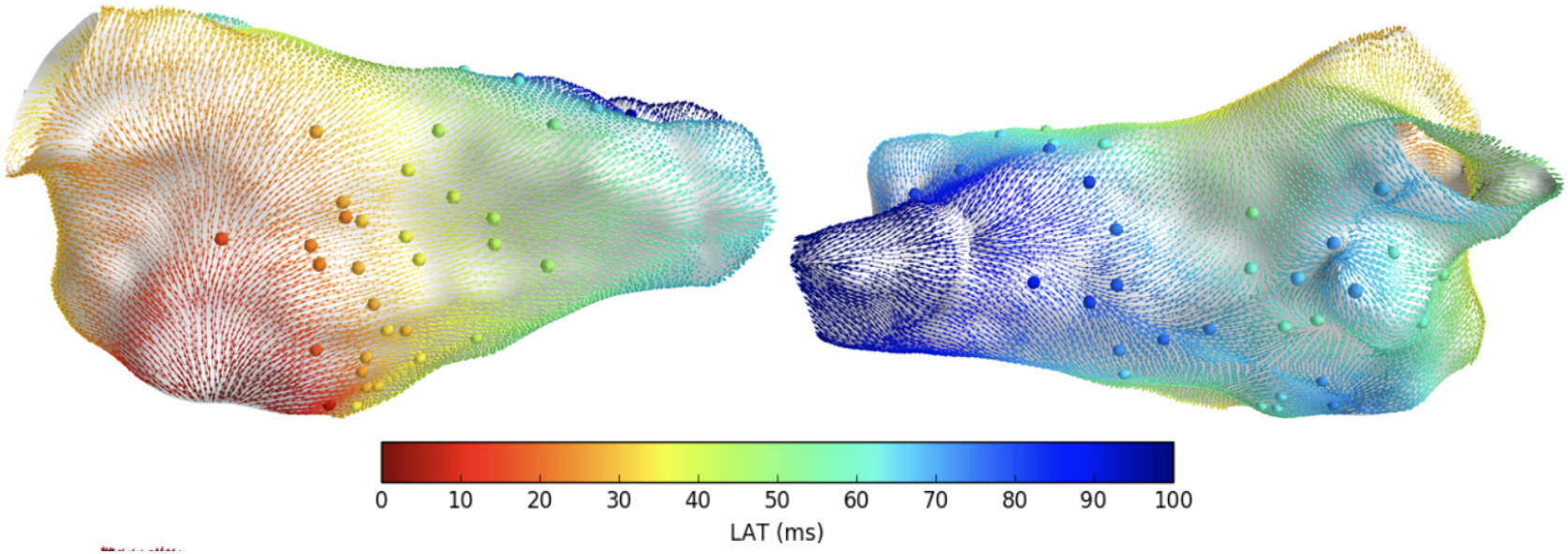
- Typically, only able to measure LAT a small number ($\sim$10) of locations on the atrium.



Interpolation between locations

We want to estimate activation times at all locations on the atria (the *LAT map*)

- Typically, only able to measure LAT a small number ($\sim$10) of locations on the atrium.



How can we interpolate to other locations?

$$LAT_{obs}(x) = \mathrm{LAT}_{true}(x) + \epsilon_{EGM} + \epsilon_{position}$$

GP interpolation

We want to model

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

but standard approaches won't work when the domain $x \in \mathcal{G}$ is an atrial manifold

- Typically covariance is a function of the Euclidean distance between two points i.e. $k(x, x') \equiv k(\|x - x'\|_2)$,

GP interpolation

We want to model

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

but standard approaches won't work when the domain $x \in \mathcal{G}$ is an atrial manifold

- Typically covariance is a function of the Euclidean distance between two points i.e. $k(x, x') \equiv k(\|x - x'\|_2)$,

We want the interpolation to take into account distance on the manifold travelled by electrical wave.

- Defining a valid positive definite covariance function on the manifold is hard!

GP basis expansions

We can consider basis expansions of GPs

$$f(x) = \sum_{i=1}^{\infty} w_i \phi_i(x)$$

where $\phi(x)$ are basis functions, and w_i random coefficients.

If $w_i \sim N(0, \lambda_i)$, then $f(x)$ is a zero-mean GP with covariance function

$$k(x, x') = \sum \lambda_i \phi_i(x) \phi_i(x')$$

GP basis expansions

We can consider basis expansions of GPs

$$f(x) = \sum_{i=1}^{\infty} w_i \phi_i(x)$$

where $\phi(x)$ are basis functions, and w_i random coefficients.

If $w_i \sim N(0, \lambda_i)$, then $f(x)$ is a zero-mean GP with covariance function

$$k(x, x') = \sum \lambda_i \phi_i(x) \phi_i(x')$$

Usually, we choose a covariance function k, and try to find convenient basis expansions

- Karhunen-Loeve expansion is mean square optimal, but inconvenient....

GP basis expansions

We can consider basis expansions of GPs

$$f(x) = \sum_{i=1}^{\infty} w_i \phi_i(x)$$

where $\phi(x)$ are basis functions, and w_i random coefficients.

If $w_i \sim N(0, \lambda_i)$, then $f(x)$ is a zero-mean GP with covariance function

$$k(x, x') = \sum \lambda_i \phi_i(x) \phi_i(x')$$

Usually, we choose a covariance function k, and try to find convenient basis expansions

- Karhunen-Loeve expansion is mean square optimal, but inconvenient....

We want to avoid specifying $k(x, x')$ explicitly, as it is difficult to do so on the atrium.

Approach 1: INLA-SPDE approach: Lindgren *et al.* 2011

Coveney *et al.* 2019

For Matern covariance functions, there is a link between GPs and stochastic partial differential equations (SPDE, Whittle) :

$$(\kappa^2 - \Delta)^{\alpha/2} f(x) = W(x)$$

Approach 1: INLA-SPDE approach: Lindgren *et al.* 2011

Coveney *et al.* 2019

For Matern covariance functions, there is a link between GPs and stochastic partial differential equations (SPDE, Whittle) :

$$(\kappa^2 - \Delta)^{\alpha/2} f(x) = W(x)$$

- Allows us to fit GPs using the machinery of finite element methods (allows solution in $O(n^{3/2})$ instead of $O(n^3)$).
- Makes it easy to work on irregular domains.

Approach 1: INLA-SPDE approach: Lindgren *et al.* 2011

Coveney *et al.* 2019

For Matern covariance functions, there is a link between GPs and stochastic partial differential equations (SPDE, Whittle) :

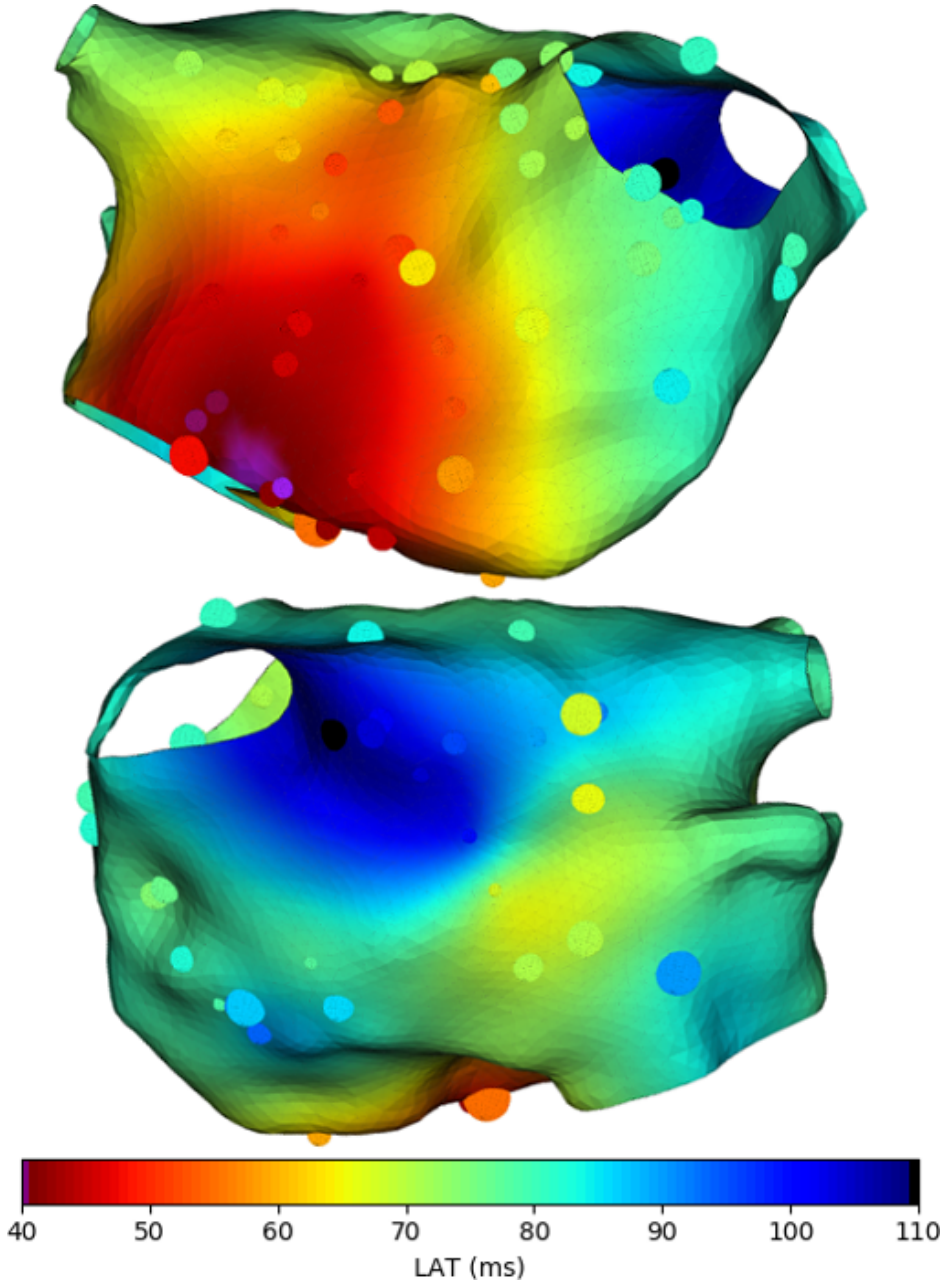
$$(\kappa^2 - \Delta)^{\alpha/2} f(x) = W(x)$$

- Allows us to fit GPs using the machinery of finite element methods (allows solution in $O(n^{3/2})$ instead of $O(n^3)$).
- Makes it easy to work on irregular domains.

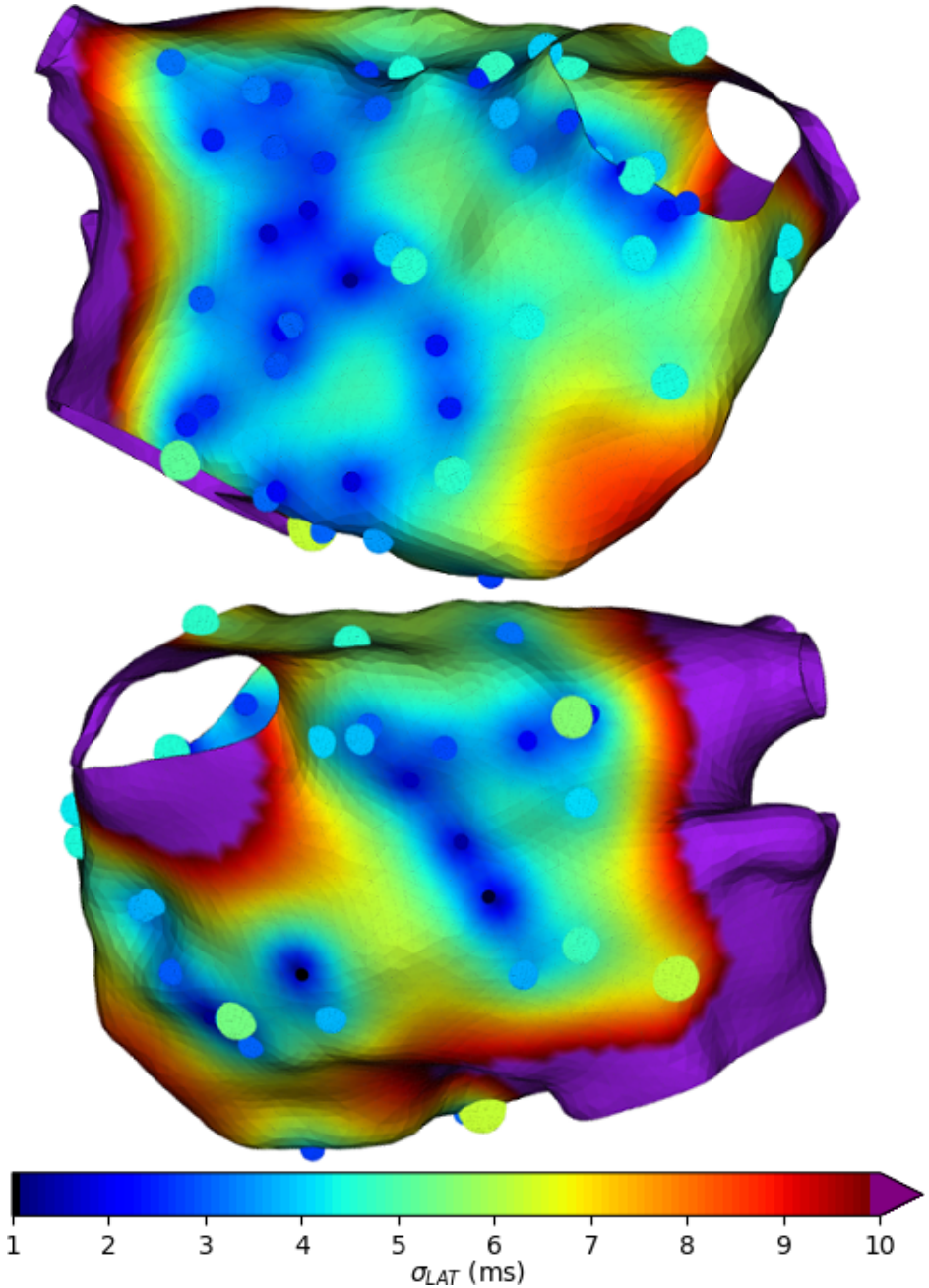
$$LAT(x) = \sum_{k=1}^{n} w_k \phi_k(x) \qquad x \in \mathcal{G}$$

with $w_k \sim N(0, \tilde{Q}^{-1})$ where $\tilde{Q}$ is sparse.

Results - mean



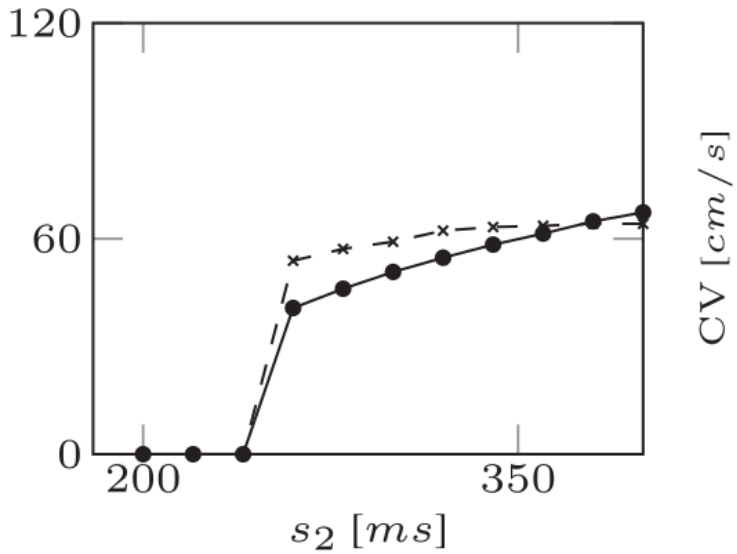
Results - standard deviation



S1-S2 interpolation

The **electrical restitution curve** describes the recovery of action potential duration as a function of the interbeat interval.

- During an EP study the heart is 'paced' at a regular S1 interval.
- Premature interbeats introduced at interval S2
- As the S2 interval shortens the heart tissue will eventually cease to recover in time to activate for both beats



S1-S2 interpolation

The EP study measures activation time at $\sim$30 locations and $\sim$ 10 S2 intervals. We use INLA-SPDE approach to interpolate LAT at the locations for a given S2 value.

- allows us to borrow strength from different S2 intervals to improve the interpolation?

S1-S2 interpolation

The EP study measures activation time at $\sim$30 locations and $\sim$ 10 S2 intervals. We use INLA-SPDE approach to interpolate LAT at the locations for a given S2 value.

- allows us to borrow strength from different S2 intervals to improve the interpolation?

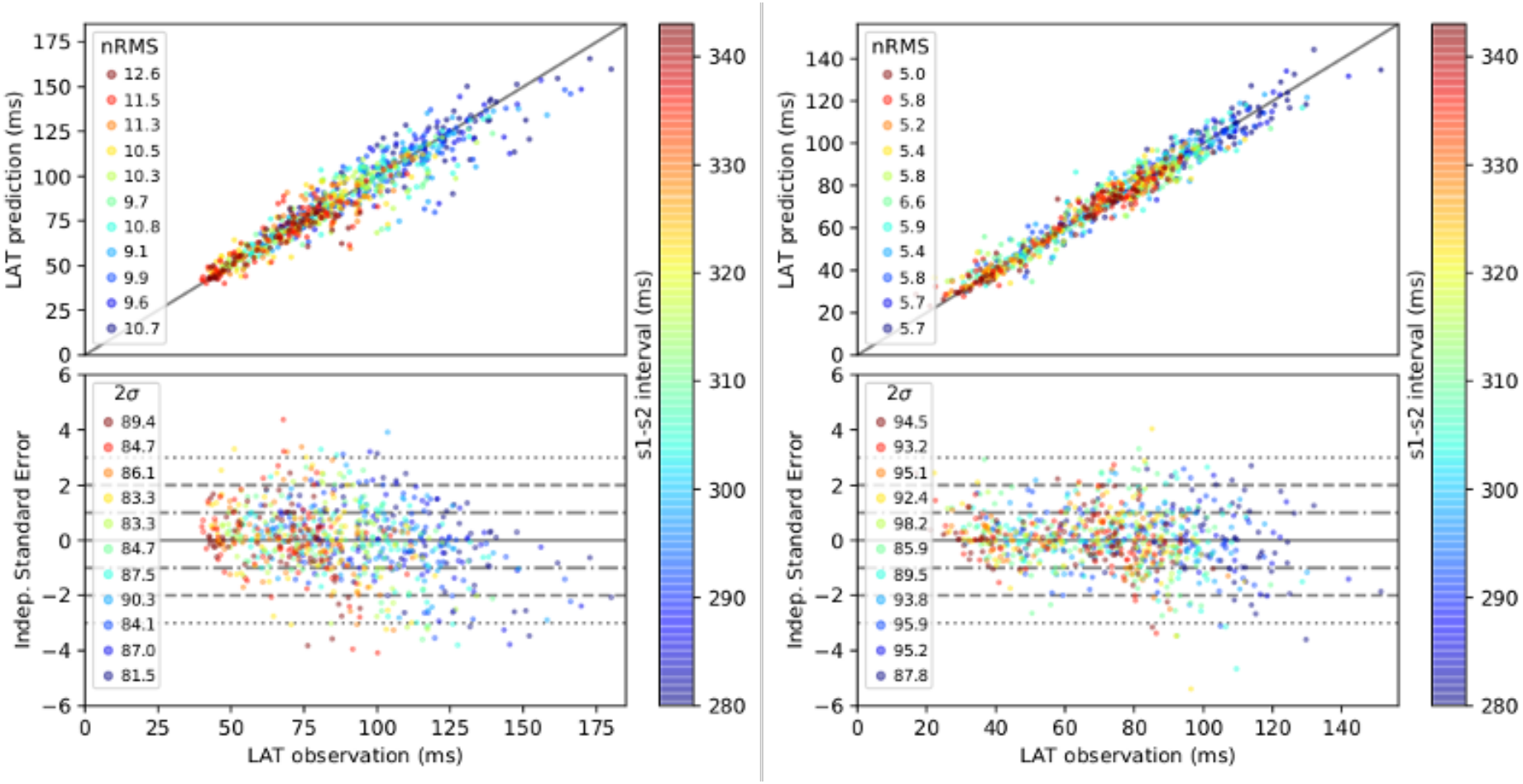
Simplest way is to add S2 as an input, and assume an AR(1) relationship between $LAT(x, S2_{i+1})$ and $LAT(x, S2_i)$

$$LAT(x, S2_{i+1}) \sim N(\rho LAT(x, S2_i), (1 - \rho^2)Q^{-1})$$

or more precisely

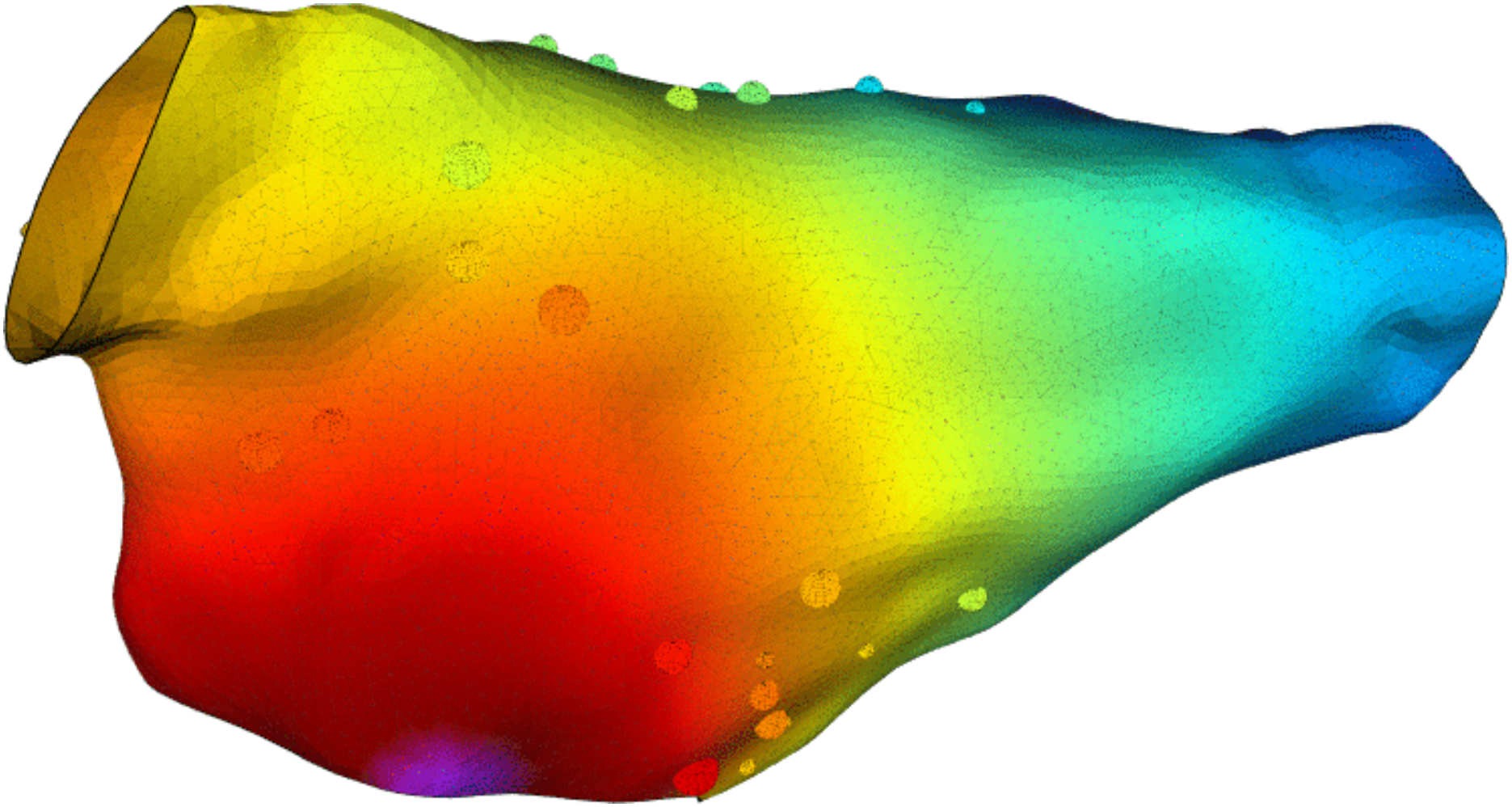
$$LAT(x, S2) \sim GP(0, Q_{S2}^{-1} \otimes Q^{-1})$$

Results: Cross validation

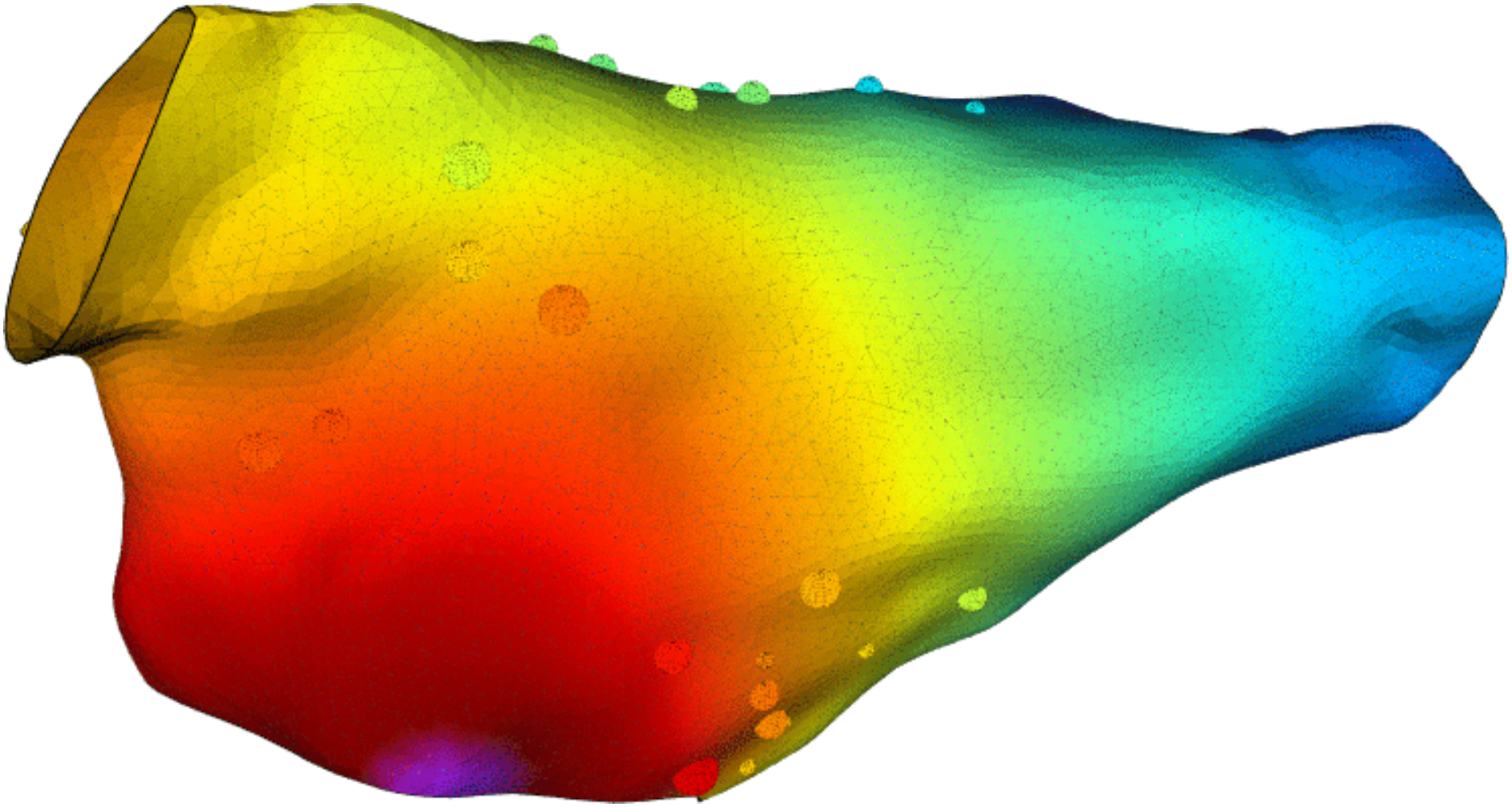


Opens interesting design questions around data collection protocols

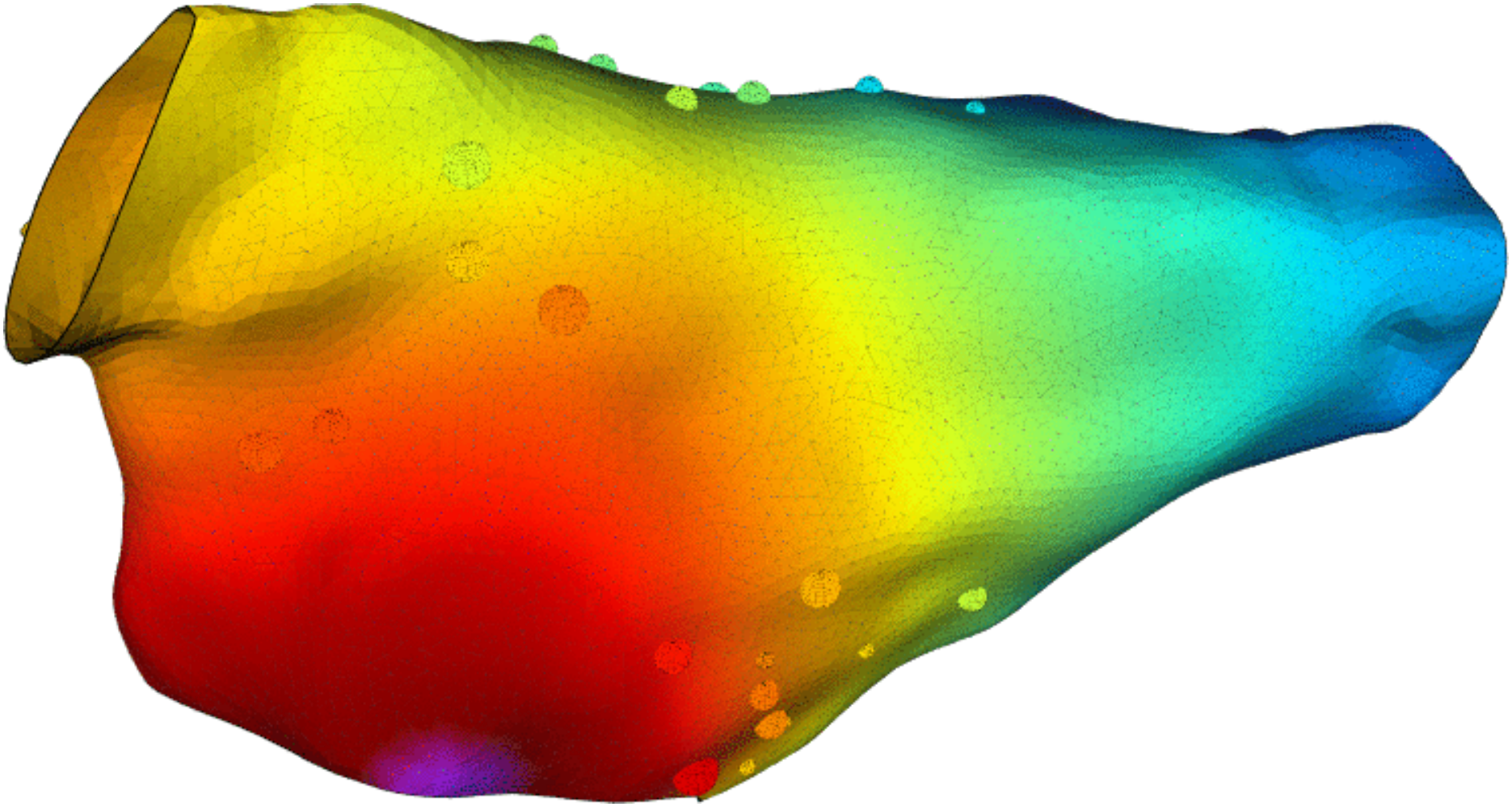
Random samples



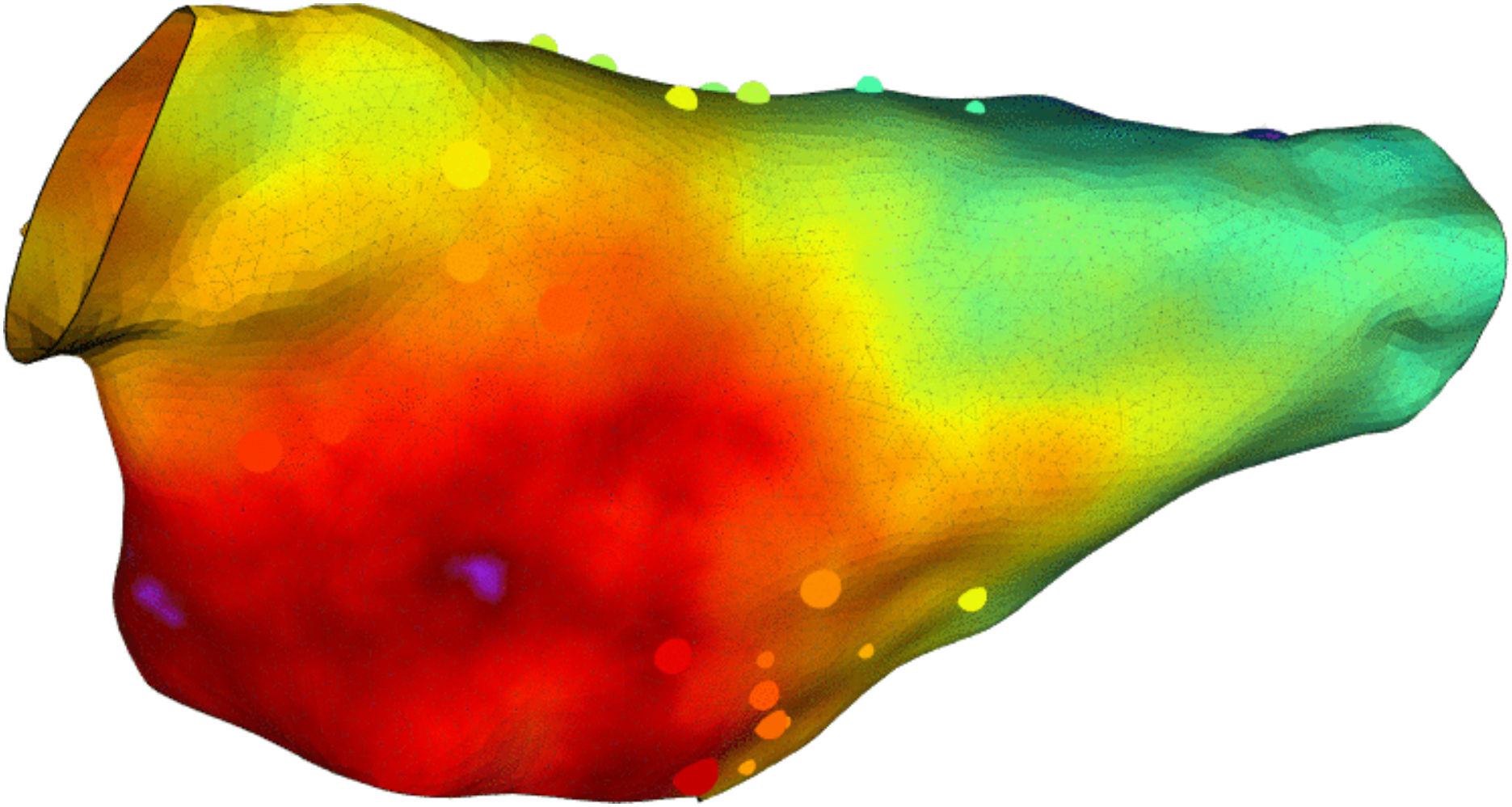
Random samples



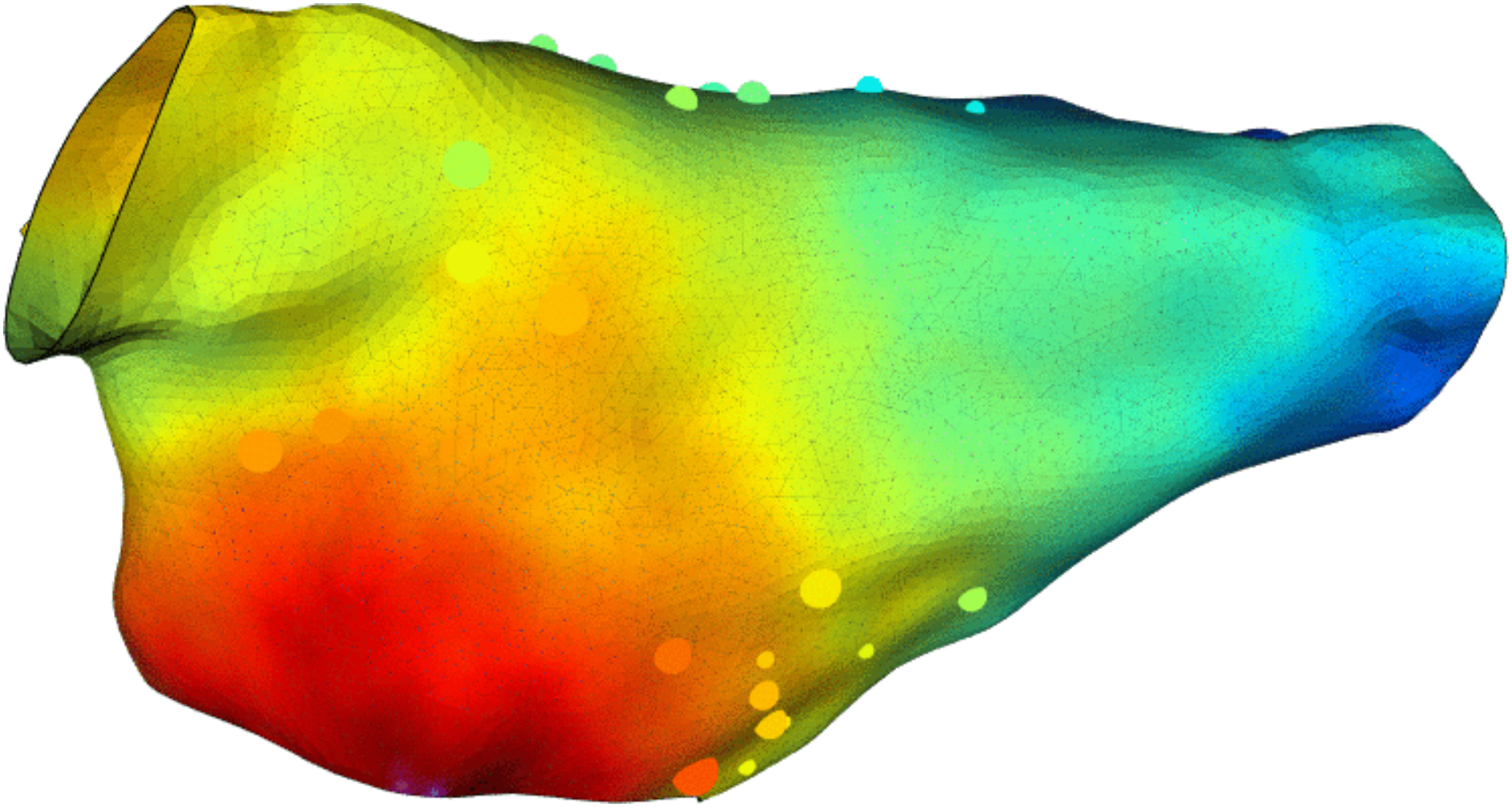
Random samples



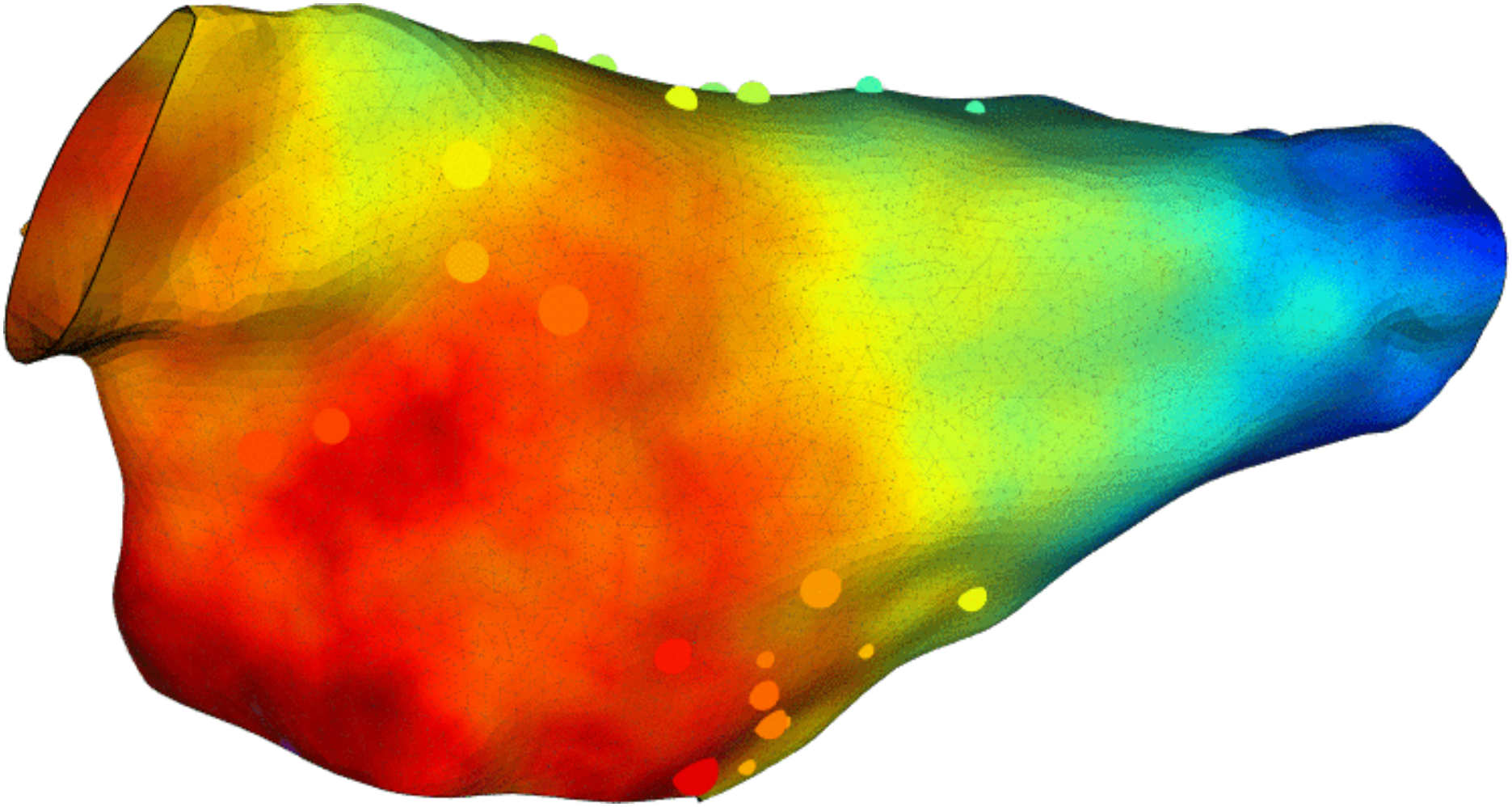
Random samples



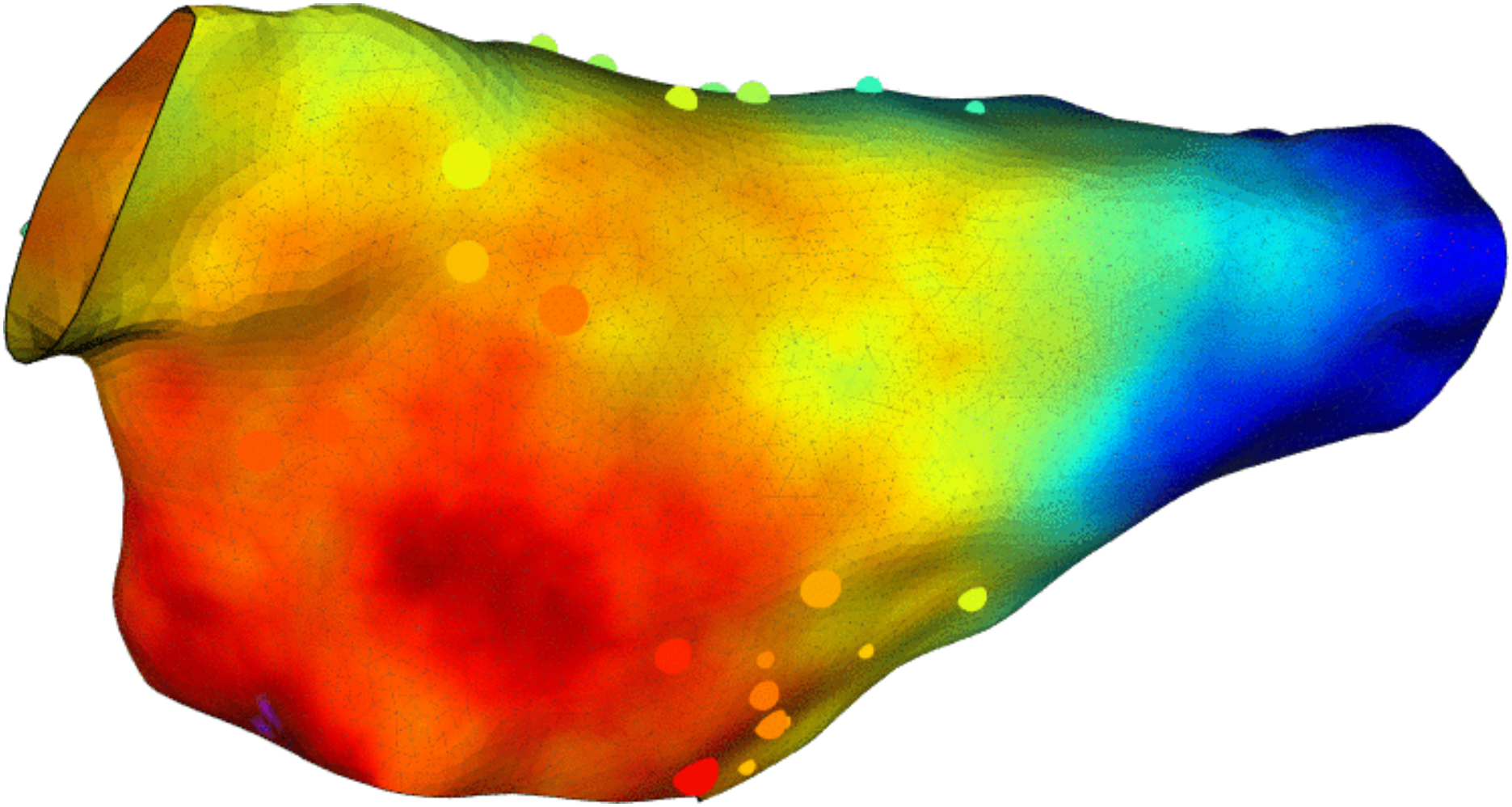
Random samples



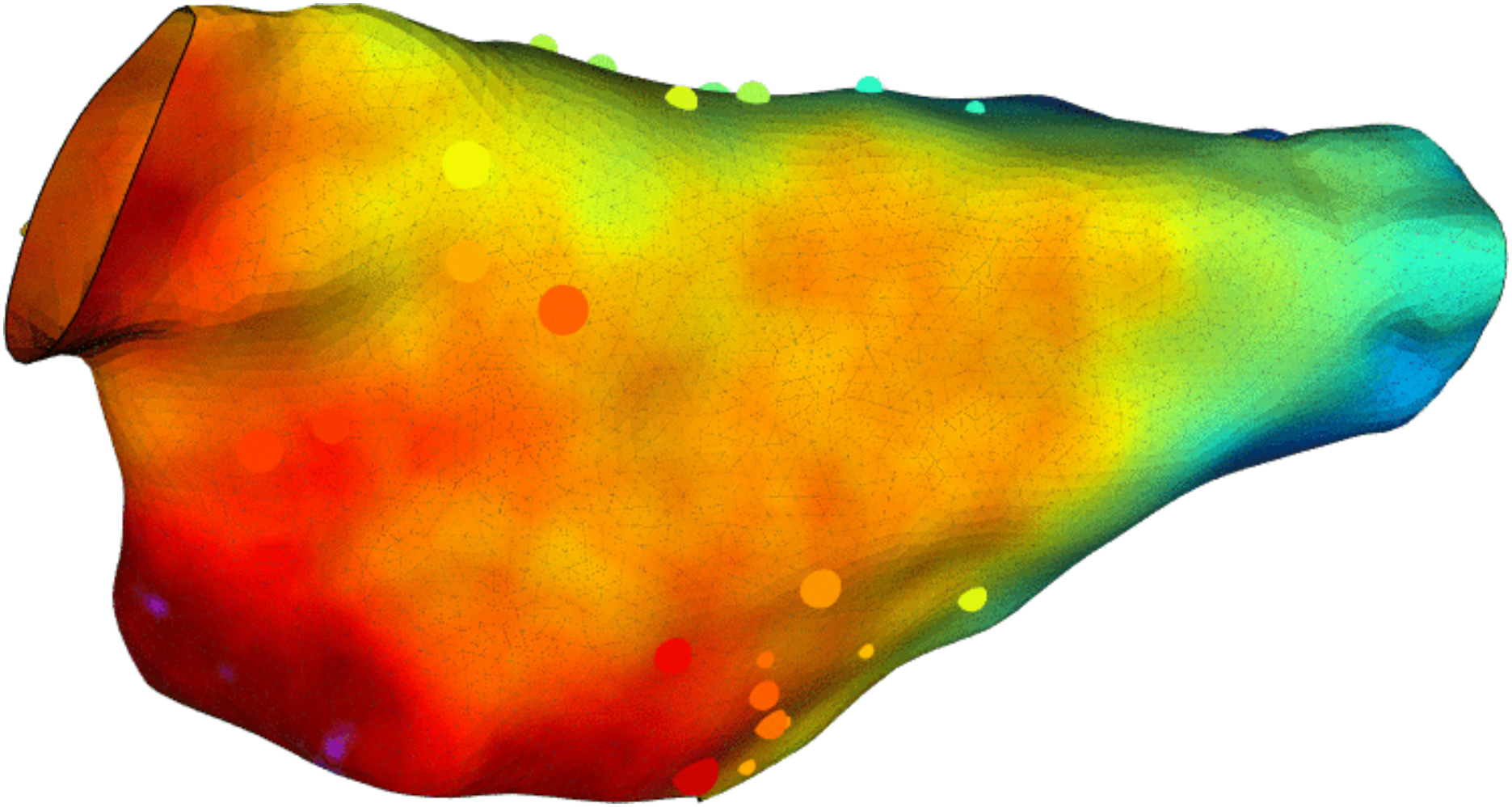
Random samples



Random samples



Random samples



Unfortunately random samples produce unphysical (non-monotonic) patterns. This isn't a surprise - the GP doesn't 'know' it is modelling a wave.

We can improve the situation by using a smoother covariance function

Approach 2: Laplacian basis functions

Coveney *et al.* Phil. Trans. Roy. Soc. 2020

There is a duality between stationary covariance functions, and spectral densities (Wiener-Khinchin):

$$S(\omega) = \int k(r) \mathrm{e}^{-i\omega r} \mathrm{d}r$$

Approach 2: Laplacian basis functions

Coveney *et al.* Phil. Trans. Roy. Soc. 2020

There is a duality between stationary covariance functions, and spectral densities (Wiener-Khinchin):

$$S(\omega) = \int k(r) \mathrm{e}^{-i\omega r} \mathrm{d}r$$

Solin and Sarkka (2019) showed that if we use the Laplacian eigenbasis

$$\begin{aligned} -\nabla^2 \phi_j(x) &= \lambda_j \phi_j(x) \qquad & x \in \mathcal{G} \\ \phi_j(x) &= 0 & x \in \partial\mathcal{G} \end{aligned}$$

then

$$f(x) = \sum w_k \phi_k(x) \qquad \text{with } w_k \sim N(0, S(\sqrt{\lambda_j}))$$

is a GP with spectral density S.

This allows us to

- specify a GP in terms of its spectral density, bypassing the need to explicitly define a covariance function
- work directly with processes on the atrial manifold

This allows us to

- specify a GP in terms of its spectral density, bypassing the need to explicitly define a covariance function
- work directly with processes on the atrial manifold

Note that

$$k(x, x') = \sum S(\sqrt{\lambda_j})\phi_i(x)\phi_i(x')$$

and that unlike many other expansions (e.g., Karhunen-Loeve), the eigenfunctions don't change if the hyper-parameters of the GP change.

This allows us to

- specify a GP in terms of its spectral density, bypassing the need to explicitly define a covariance function
- work directly with processes on the atrial manifold

Note that

$$k(x, x') = \sum S(\sqrt{\lambda_j})\phi_i(x)\phi_i(x')$$

and that unlike many other expansions (e.g., Karhunen-Loeve), the eigenfunctions don't change if the hyper-parameters of the GP change. Truncating the sum gives us an approximate low rank GP

$$k(x, x') \approx \sum_{i=1}^{M} S(\sqrt{\lambda_j})\phi_i(x)\phi_i(x'), \quad f(x) \approx \sum_{i=1}^{M} w_k\phi_k(x)$$

for which inference can be done in $O(M^3)$ operations.

Computing conduction velocities

Interest lies in conduction velocities, which are the inverse of the LAT gradient. The Laplacian eigen expansion allows us to compute these

$$\mathbb{E}\left[\frac{\partial f(\mathbf{x}^*)}{\partial \mathbf{x}^*} \mid \mathcal{D}\right] = \frac{\partial \mathbf{k}_*^{\mathbf{T}}}{\partial \mathbf{x}^*}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\mathbf{y}$$

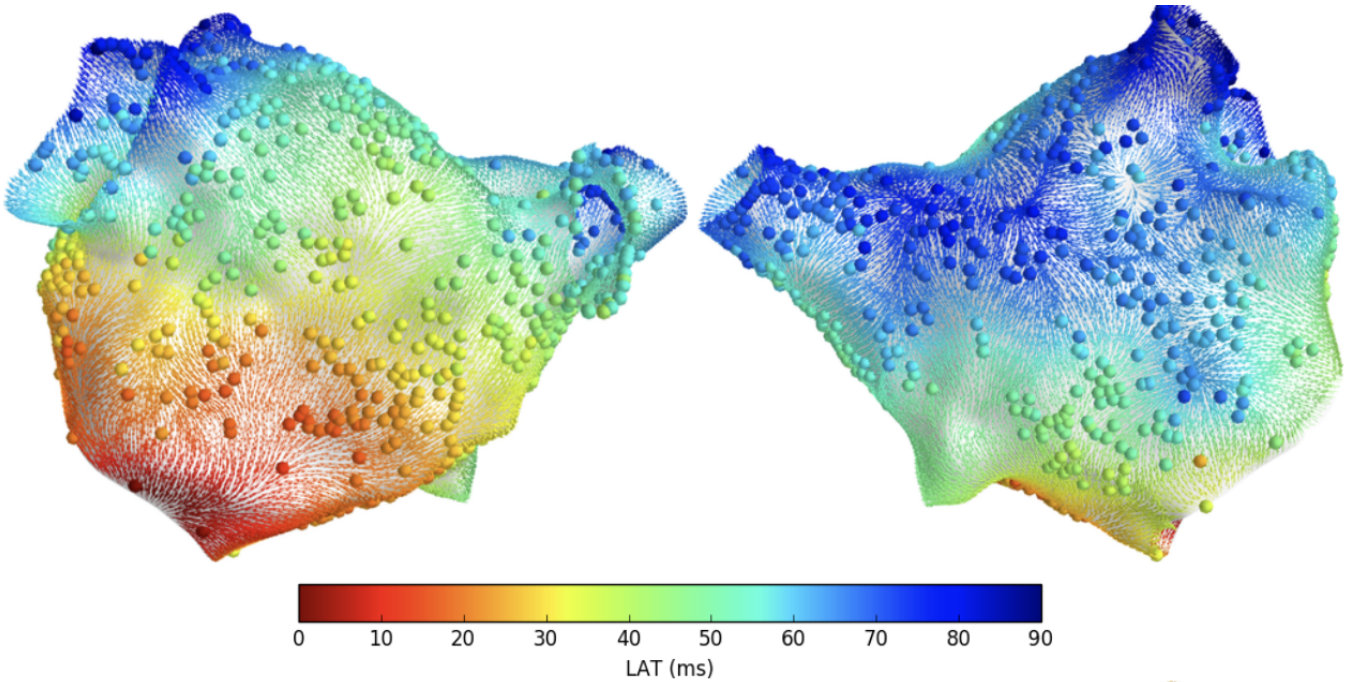
$$\mathbb{V}\left[\frac{\partial f(\mathbf{x}^*)}{\partial \mathbf{x}^*} \mid \mathcal{D}\right] = \tau^2 \left.\frac{\partial^2 k(\mathbf{x_a}, \mathbf{x_b})}{\partial \mathbf{x_a} \partial \mathbf{x_b}}\right|_{\mathbf{x_a}=\mathbf{x_b}=\mathbf{x}^*} - \frac{\partial \mathbf{k}_*^{\mathbf{T}}}{\partial \mathbf{x}^*}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\frac{\partial \mathbf{k}_*}{\partial \mathbf{x}^*}$$

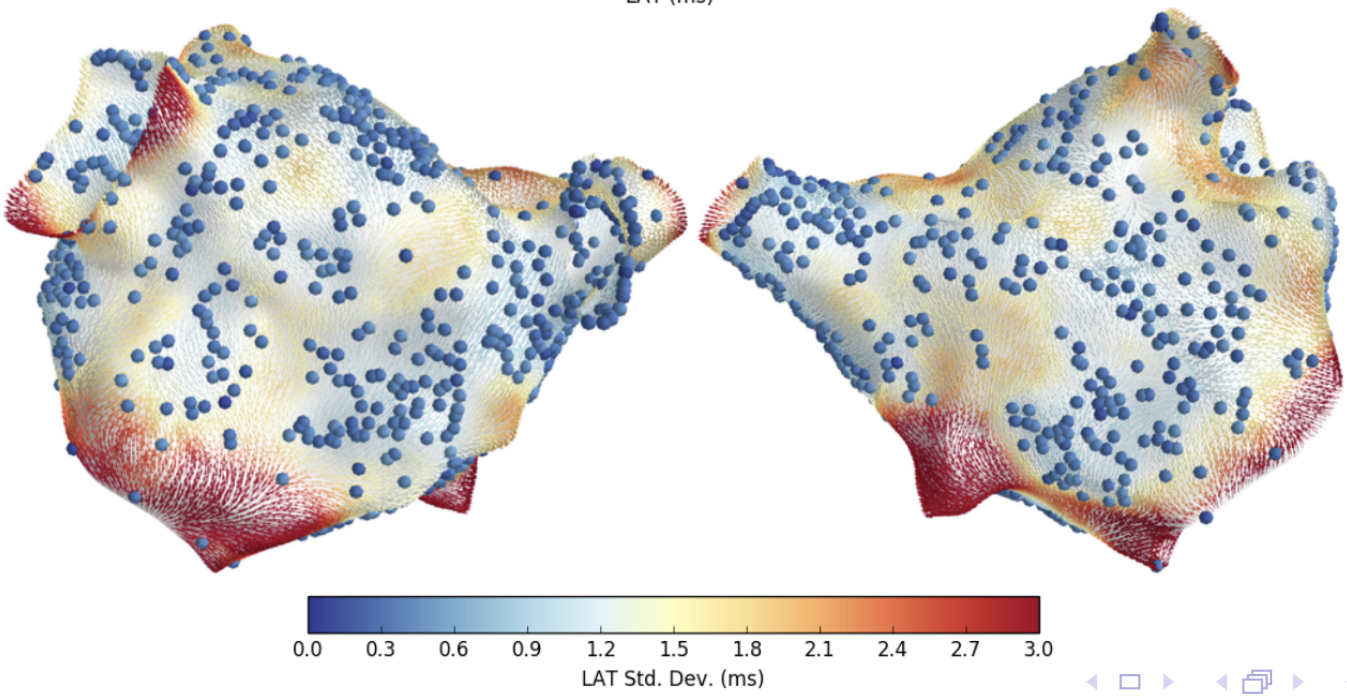
where

$$\frac{dk(x,x')}{dx} = \sum_{i=1}^{M} S(\sqrt{\lambda_j})\frac{d\phi_i}{dx}(x)\phi_i(x')$$

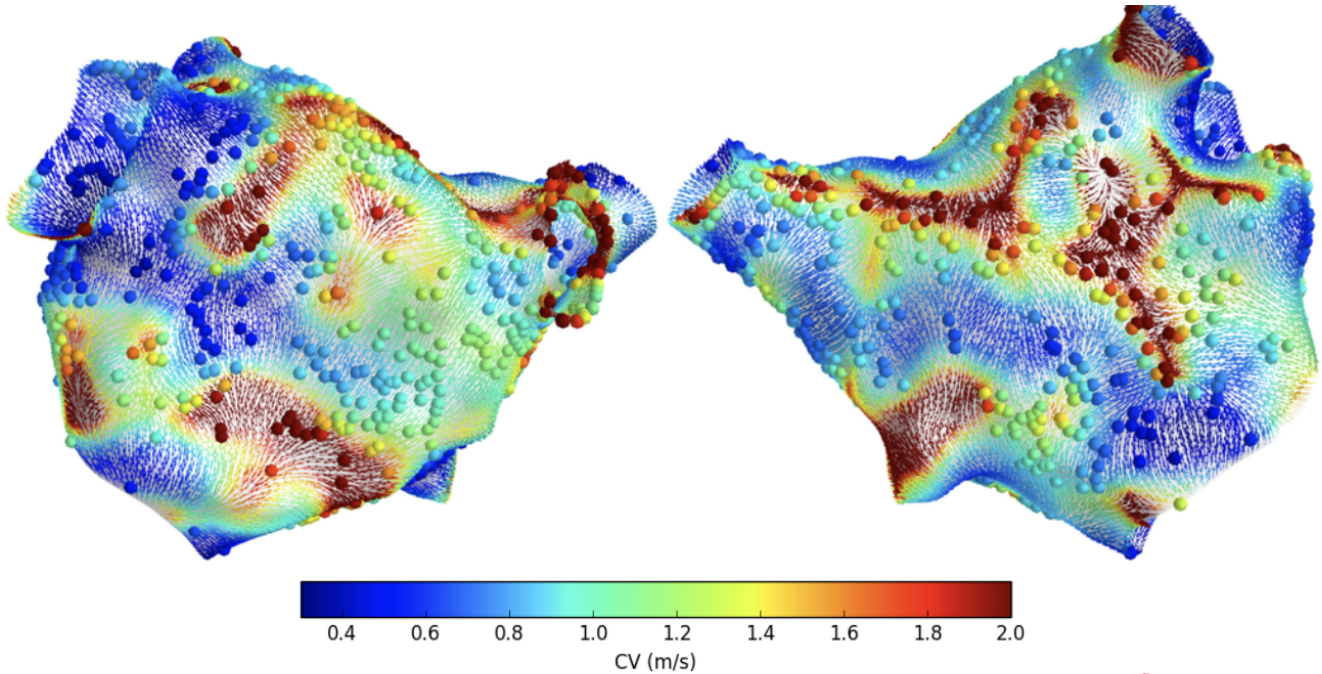
allowing us to compute variance estimates of the estimated conduction velocities...

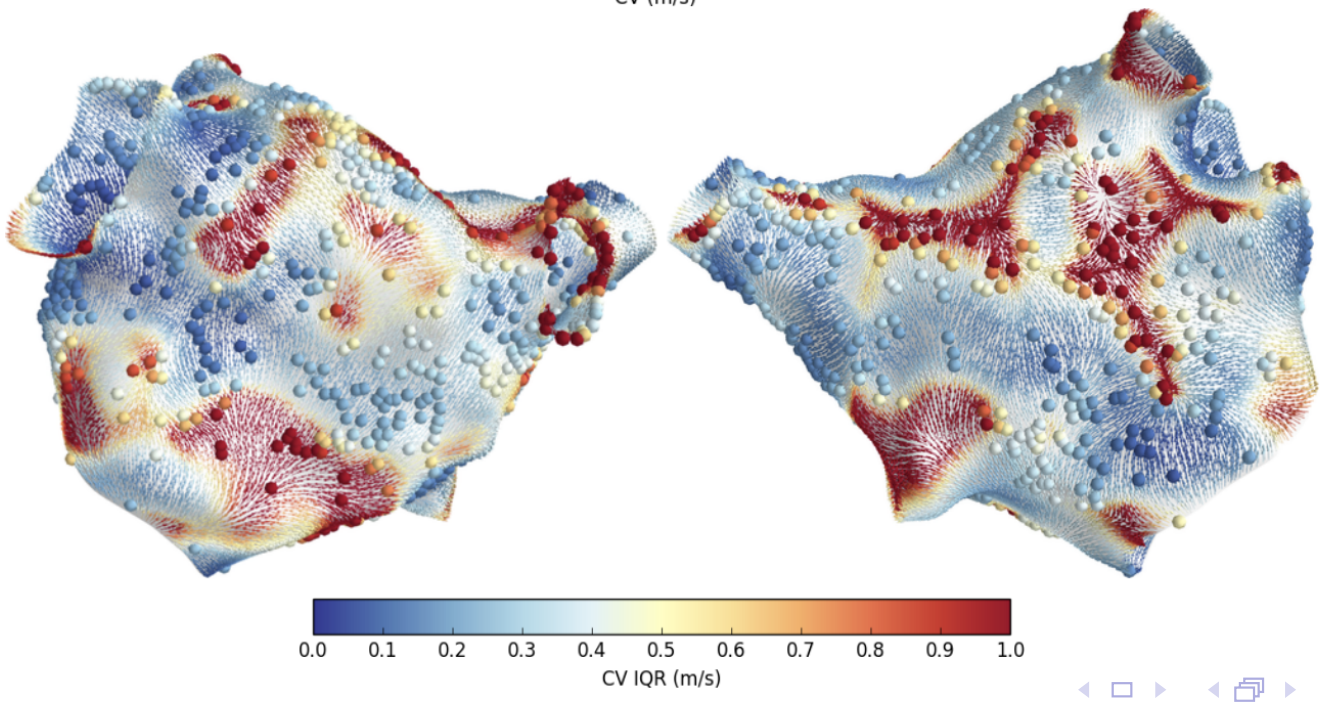
Results


0
10
20
30
40
50
60
70
80
90
LAT (ms)


0.0
0.3
0.6
0.9
1.2
1.5
1.8
2.1
2.4
2.7
3.0
LAT Std. Dev. (ms)

Results


0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
CV (m/s)

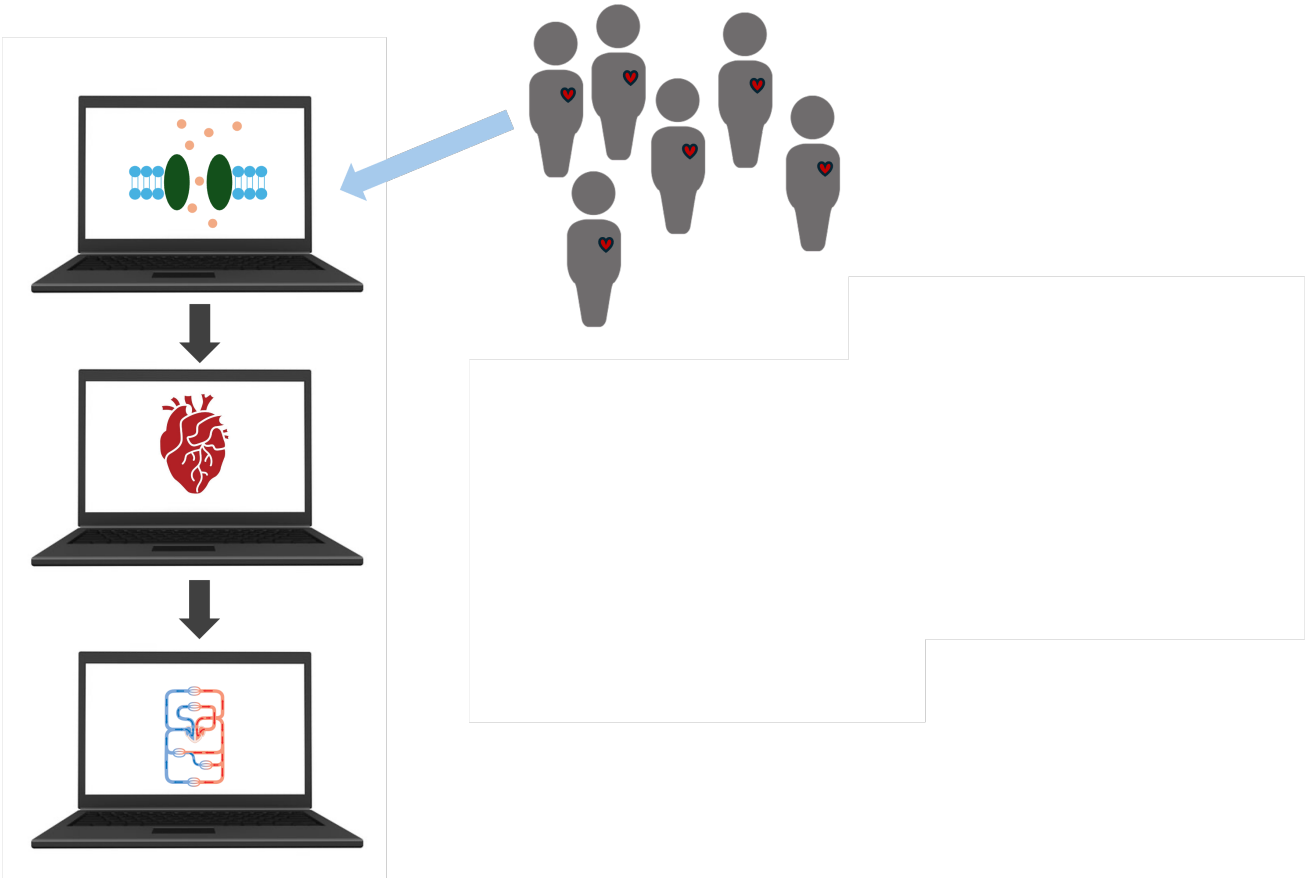

0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
CV IQR (m/s)

Problem 2: Cohort emulation

Chris Lanyon

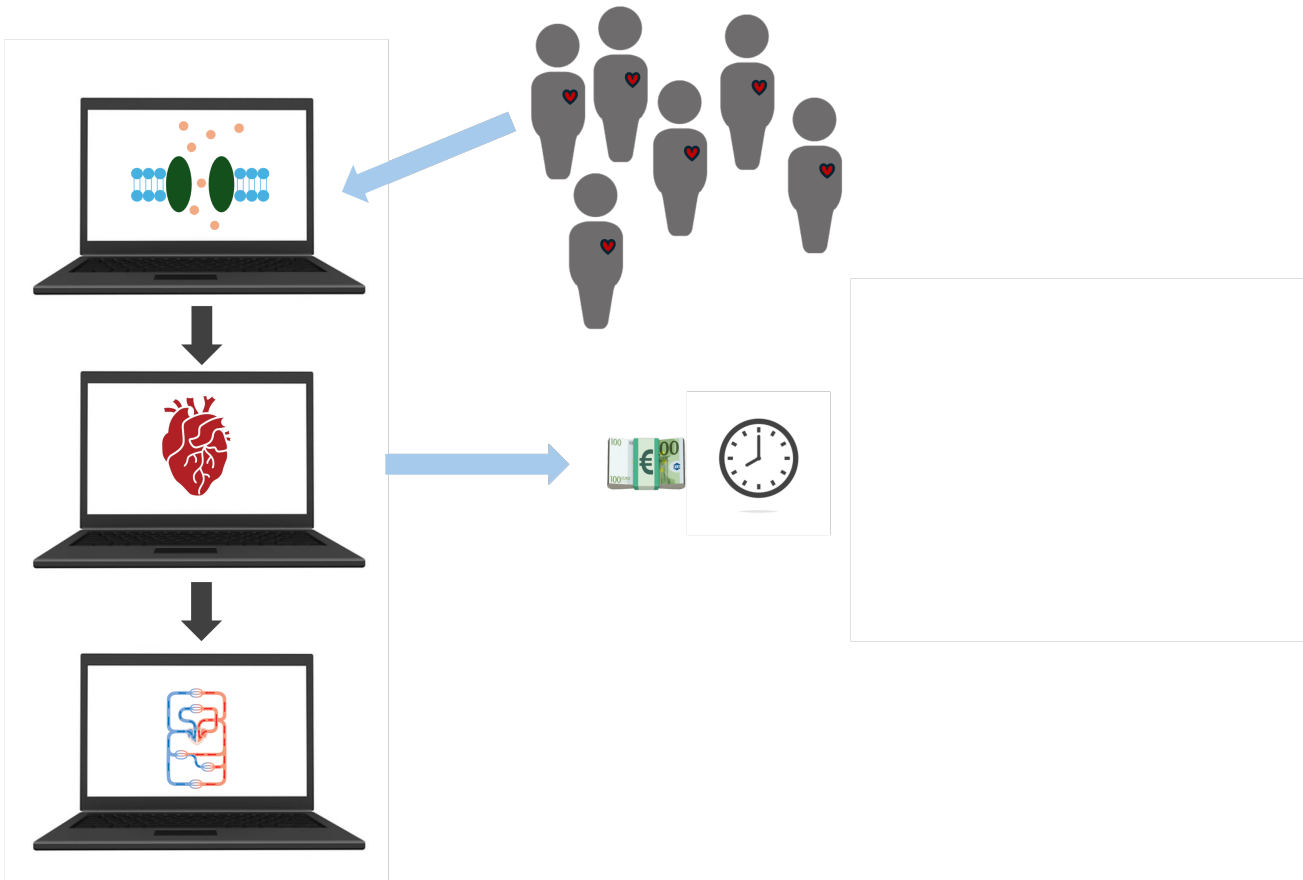
Problem 2: Cohort emulation

Chris Lanyon



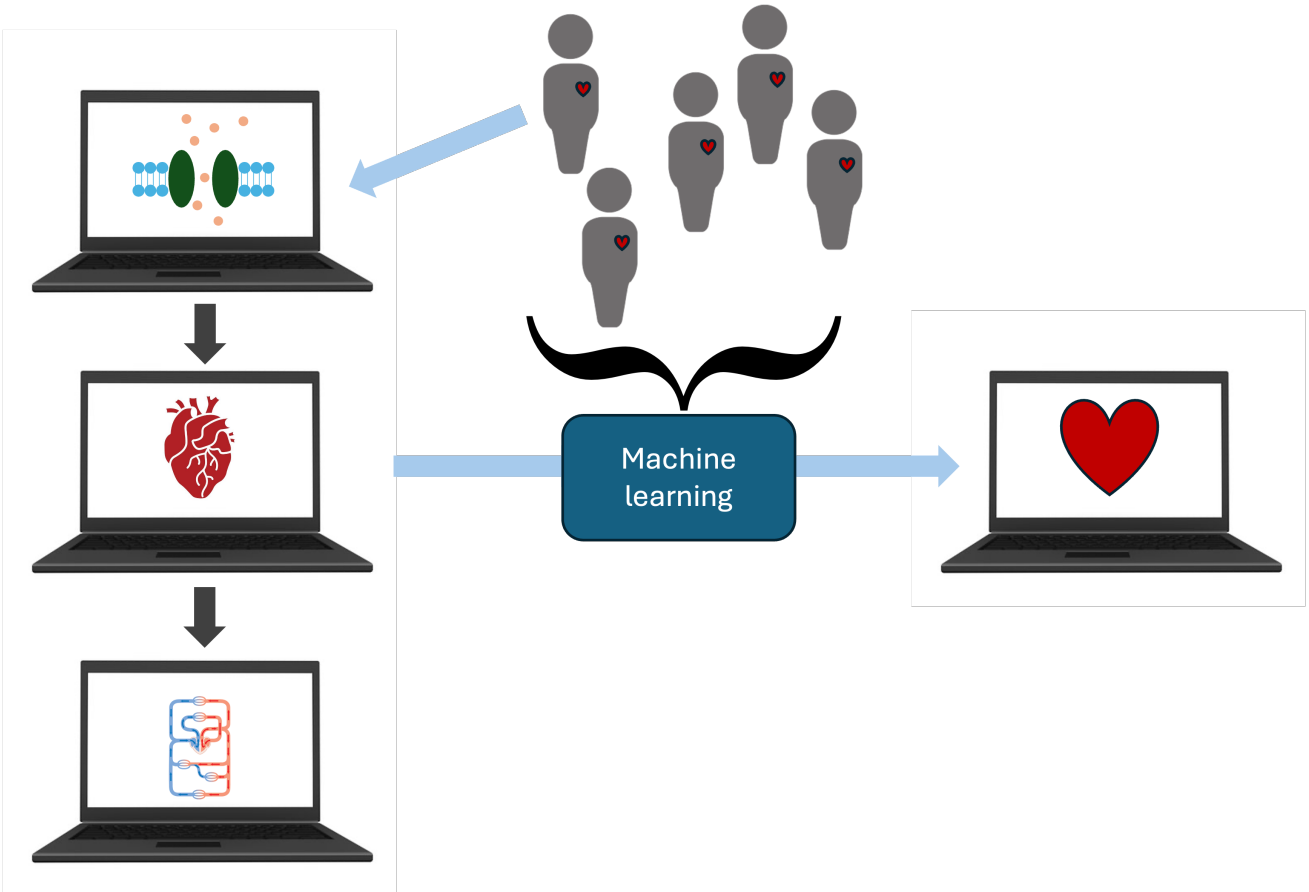
Problem 2: Cohort emulation

Chris Lanyon



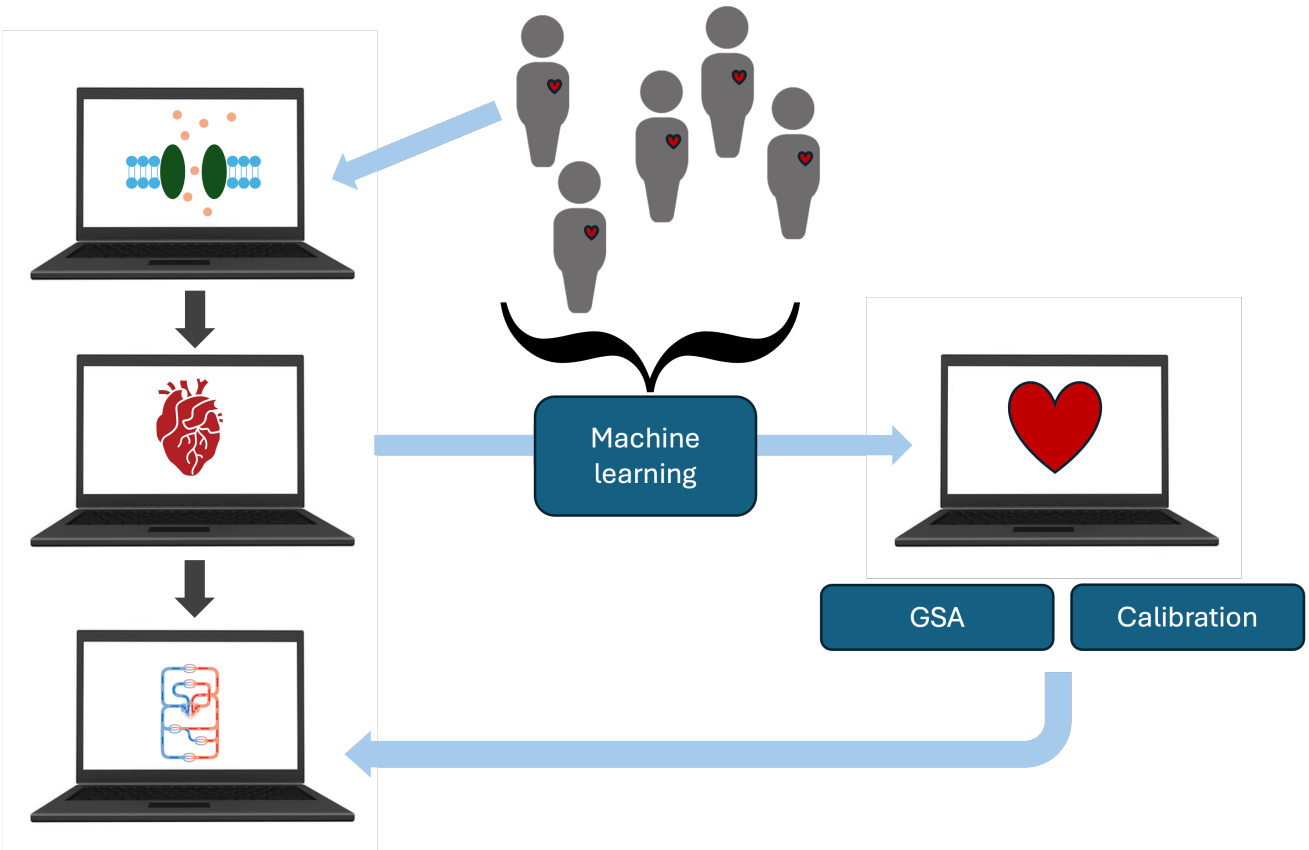
Problem 2: Cohort emulation

Chris Lanyon



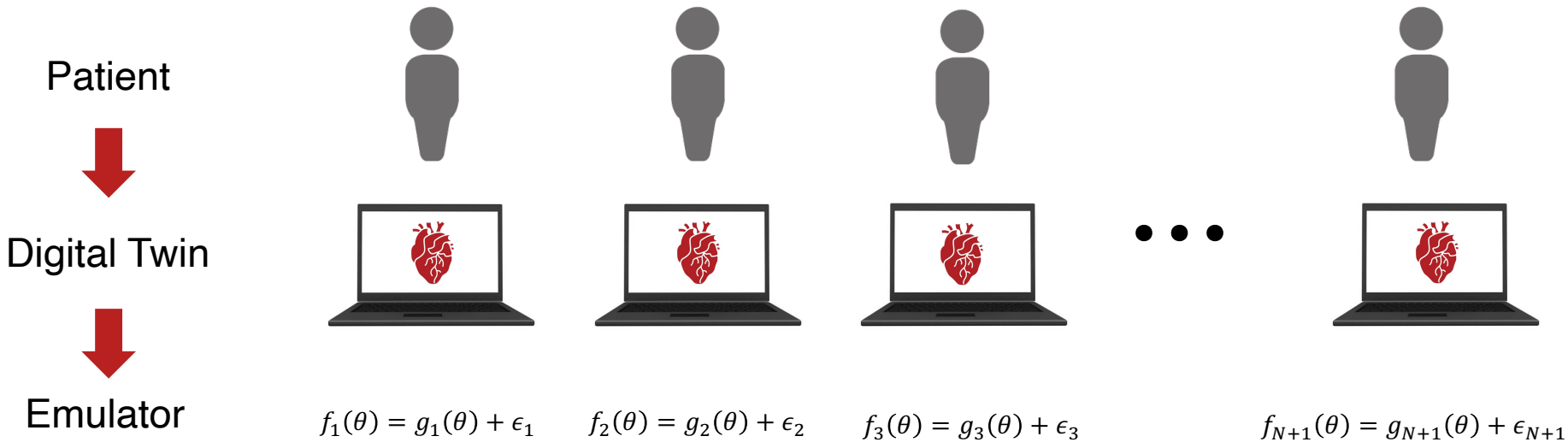
Problem 2: Cohort emulation

Chris Lanyon



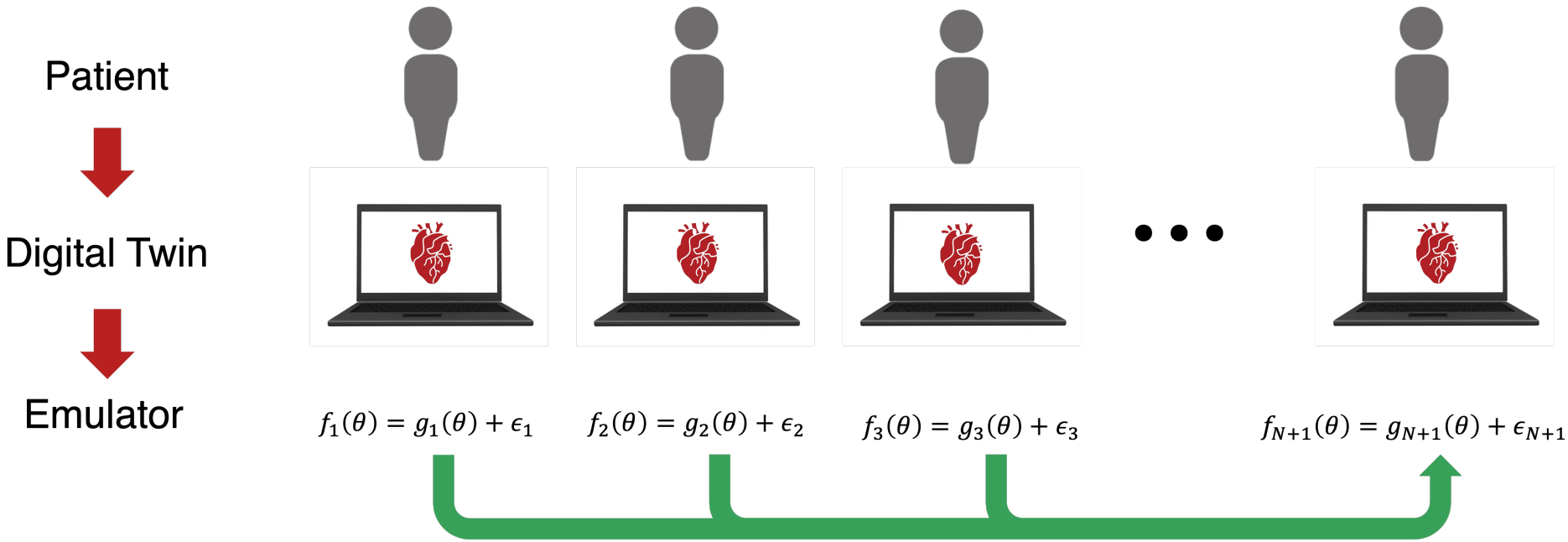
Leveraging the Digital Tapestry: Cohort emulation

- Our main cost (temporal, financial, computational) is forward runs of each patient's digital twin
- Emulators can run the forward model quickly but require simulator runs for training
- Default approach is a single emulator per cohort member: $f_i(\theta) = g_i(\theta) + \epsilon_i$



Leveraging the Digital Tapestry: Cohort emulation

- **Our aim is to leverage information from the cohort to reduce the computational cost of building emulators**



Cohort learning method 1: Discrepancy emulators

Intuition: Say we gain new patients sequentially, we've learned a lot about our first patient's heart, what do we know about the next patient?
Propose a discrepancy model:

$$f_1(\theta) \approx g_1(\theta) = ag_0(\theta) + \delta(\theta)$$

Cohort learning method 1: Discrepancy emulators

Intuition: Say we gain new patients sequentially, we've learned a lot about our first patient's heart, what do we know about the next patient?
Propose a discrepancy model:

$$f_1(\theta) \approx g_1(\theta) = a g_0(\theta) + \delta(\theta)$$

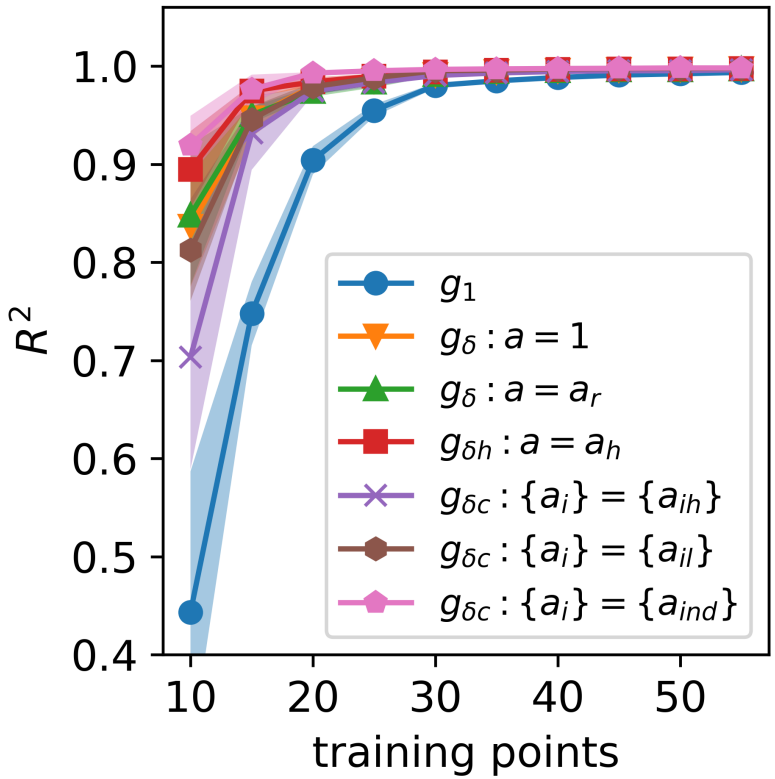
We can extend this approach to leverage information from the full cohort of models:

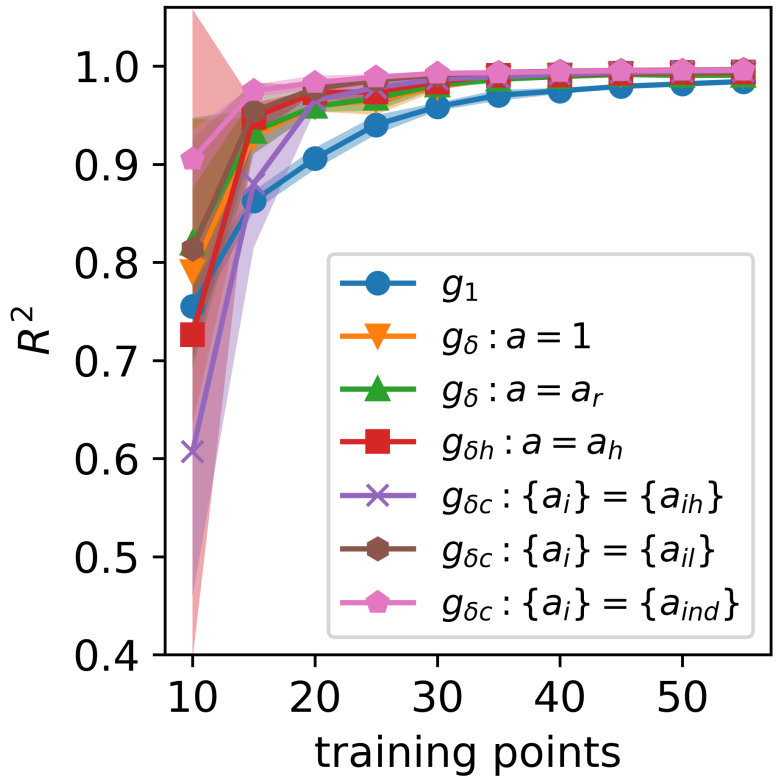
$$f_{N+1}(\theta) \approx g_{N+1}(\theta) = \sum_{i=1}^{N} a_i g_i(\theta) + \delta(\theta)$$

Due to the additive property of GPs $g_{N+1}(\theta)$ is also a GP and can be trained using GP regression.

Results: Discrepancy emulators

Rodero *et al.* 2021, Baptiste *et al.* 2025, Lanyon *et al.* later in 2025





Cohort learning method 2: Latent feature emulators

Intuition: Our data generating function for each patient, $f_i(\theta)$, is actually a function over each patient's cardiac geometry, G, such that

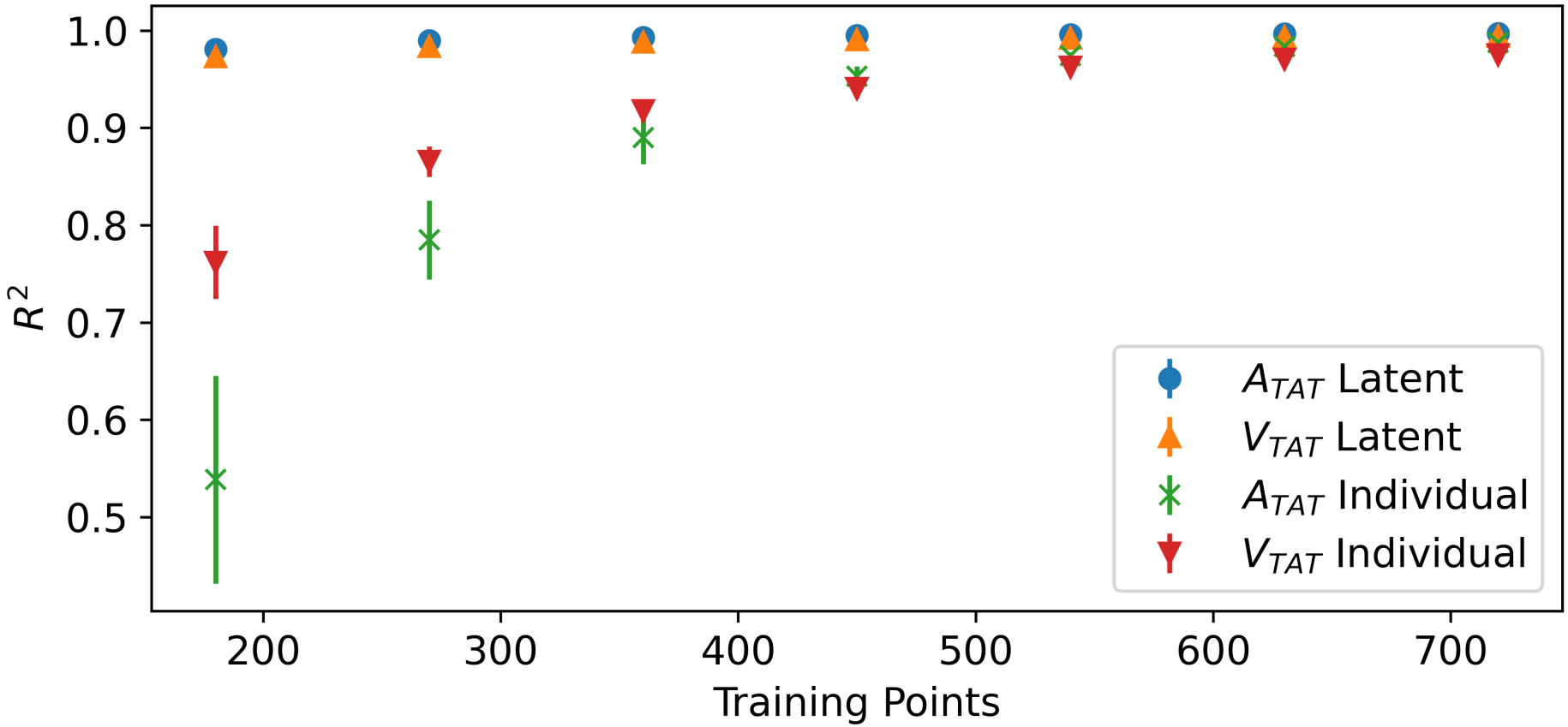
$$f_i(\theta) = f(\theta, G_i)$$

On this basis we aim to learn a latent space representation of the geometry, I, and use it to train a single Gaussian process emulator, $g(\theta, I)$ such that

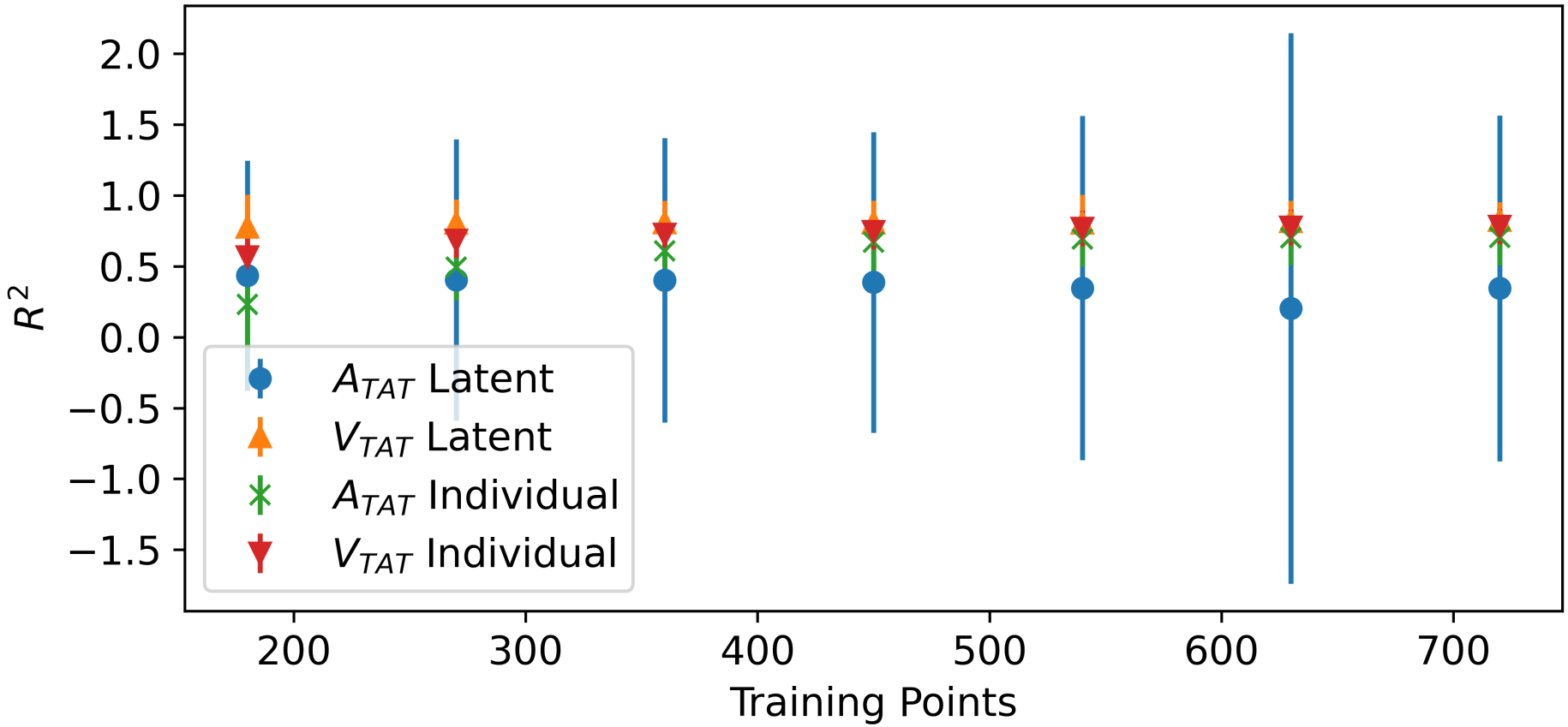
$$f(\theta, G_i) = g(\theta, I_i) + \epsilon$$

- Latent features can either be derived from a shape model or be arbitrary (cf GP-LVM).

Results: Latent emulators, left in meshes



Results: Latent emulators, left out meshes



Problem 3: Estimating tissue parameters

We have

- Physics based electrophysiology simulator $f(\theta)$ that models LAT given (spatially varying) tissue parameters $\theta(x)$ for $x \in \mathcal{G}$.
- Emulators that approximate the simulator at low cost
- Observations, y, of LAT collected in EP studies

Problem 3: Estimating tissue parameters

We have

- Physics based electrophysiology simulator $f(\theta)$ that models LAT given (spatially varying) tissue parameters $\theta(x)$ for $x \in \mathcal{G}$.
- Emulators that approximate the simulator at low cost
- Observations, y, of LAT collected in EP studies

We need to

- estimate the parameters from the EP data $\pi(\theta(\cdot)|y)$
- and predict the result of ablation therapy.

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

during a 30min procedure!

Parameter estimation

Local approach

- At each location x_i, infer $\theta(x_i)$ using ABC with a look-up table of simulations
- Interpolate $\theta(x)$ across the atrium.

Parameter estimation

Local approach

- At each location x_i, infer $\theta(x_i)$ using ABC with a look-up table of simulations
- Interpolate $\theta(x)$ across the atrium.

Dimension reduction: Find a projection $P : \mathbb{R}^D \to \mathbb{R}^d$ to project the parameter into a lower dimensional space and model low dimensional approximation to the simulator:

$$f'(z) = f(P^\top z)$$

Parameter estimation

Local approach

- At each location x_i, infer $\theta(x_i)$ using ABC with a look-up table of simulations
- Interpolate $\theta(x)$ across the atrium.

Dimension reduction: Find a projection $P : \mathbb{R}^D \to \mathbb{R}^d$ to project the parameter into a lower dimensional space and model low dimensional approximation to the simulator:

$$f'(z) = f(P^\top z)$$

We can choose P by

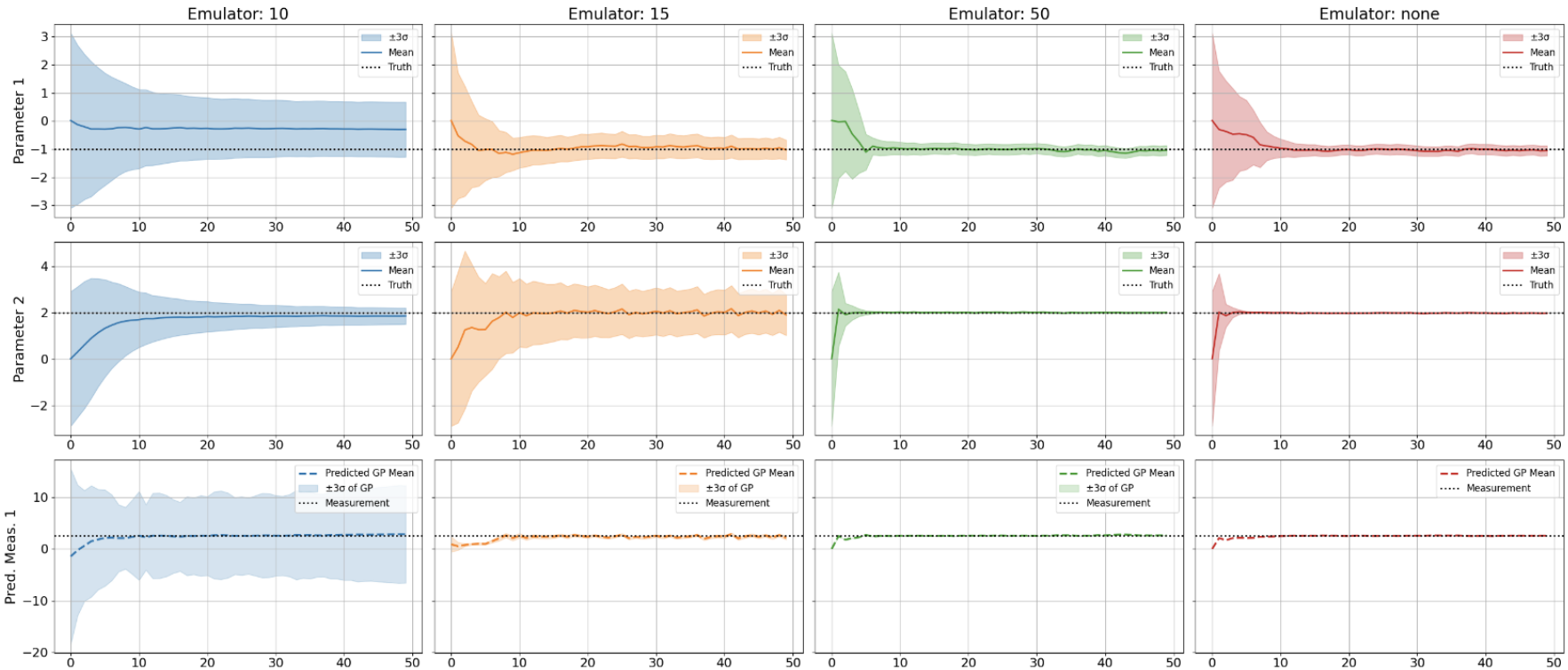
- Global sensitivity analysis
- Active subspace methods
- ML magic: embedding learning methods, VAE etc
- Handcrafted projections

Real time calibration with the Ensemble Kalman Filter

Mamajiwala *et al.* 2024, 2025 (forthcoming)

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

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



Conclusions

- We can currently build DTs for a single patient, but at great expense
  - Need to scale and speed up this process
- We need:
  - to find regularities in the problem to allow us to reduce dimension
  - to learn strong population structured prior distributions
  - to develop fast method to approximately infer parameters.
- Gaussian processes have become a key part of the DT pipeline!

Conclusions

- We can currently build DTs for a single patient, but at great expense
  - Need to scale and speed up this process
- We need:
  - to find regularities in the problem to allow us to reduce dimension
  - to learn strong population structured prior distributions
  - to develop fast method to approximately infer parameters.
- Gaussian processes have become a key part of the DT pipeline!

Thank you for listening!

See `cvd-net.com/` for job opportunities: