

# Using flexible noise models to avoid noise model misspecification in inference of differential equation time series models

Richard Creswell

*Department of Computer Science, University of Oxford, Oxford, United Kingdom*

E-mail: richard.creswell@hertford.ox.ac.uk

Ben Lambert

*MRC Centre for Global Infectious Disease Analysis, School of Public Health, Imperial College London, United Kingdom*

Chon Lok Lei

*Department of Computer Science, University of Oxford, Oxford, United Kingdom*

Martin Robinson

*Department of Computer Science, University of Oxford, Oxford, United Kingdom*

David Gavaghan

*Department of Computer Science, University of Oxford, Oxford, United Kingdom*

**Summary.** When modelling time series, it is common to decompose observed variation into a “signal” process, the process of interest, and “noise”, representing nuisance factors that obfuscate the signal. To separate signal from noise, assumptions must be made about both parts of the system. If the signal process is incorrectly specified, our predictions using this model may generalise poorly; similarly, if the noise process is incorrectly specified, we can attribute too much or too little observed variation to the signal. With little justification, independent Gaussian noise is typically chosen, which defines a statistical model that is simple to implement but often misstates system uncertainty and may underestimate error autocorrelation. There are a range of alternative noise processes available but, in practice, none of these may be entirely appropriate, as actual noise may be better characterised as a time-varying mixture of these various types. Here, we consider systems where the signal is modelled with ordinary differential equations and present classes of flexible noise processes that adapt to a system’s characteristics. Our noise models include a multivariate normal kernel where Gaussian processes allow for non-stationary persistence and variance, and nonparametric Bayesian models that partition time series into distinct blocks of separate noise structures. Across the scenarios we consider, these noise processes faithfully reproduce true system uncertainty: that is, parameter estimate uncertainty when doing inference using the correct noise model. The models themselves and the methods for fitting them are scalable to large datasets and could help to ensure more appropriate quantification of uncertainty in a host of time series models.

*Keywords:* time series, ordinary differential equations, noise model, non-stationary processes, Bayesian inference, Gaussian process, Bayesian nonparametrics

## 1. Introduction

Time series data are ubiquitous in many fields of scientific inquiry. In this paper, we focus on time series data that are assumed to obey a (potentially nonlinear) parametric model  $f(t; \theta)$ , a function of time  $t$  and model parameters  $\theta$ . We model a noise-free trajectory  $\{\bar{y}_i\}_{i=1}^N$  at time points  $\{t_i\}_{i=1}^N$  according to,

$$\bar{y}_i = f(t_i; \theta). \quad (1)$$

Often, scientific knowledge about a given system does not specify  $f$  directly but rather suggests an ordinary differential equation (ODE) or system of ODEs with  $f$  as their solution. In the following, we assume that these equations are numerically solvable, so that the mean or noise-free trajectory<sup>†</sup>  $f(t; \theta)$  is readily available for given values of  $t$  and  $\theta$ : this is known as solving the “forward problem”. Here, we consider the so-called “inverse problem”: where, given a time series of noisy observations, the aim is to infer the values of  $\theta$  that have generated the observations. A wide variety of inference tasks fall into this category, including parameter estimation for enzyme kinetics, biochemical pathways, and regulatory dynamics (Milstein, 1981; Moles et al., 2003; Silk et al., 2011). The Bayesian approach to the inverse problem, which we adopt in this paper, yields a posterior probability distribution over model parameters that conveys uncertainty in parameter estimates implied by data (Gelman et al., 2013).

When solving the forward problem, assumptions made about the form of the noise can substantially change estimated posterior uncertainty of  $\theta$  (Lambert et al., 2020). Notably, when the noise model is misspecified, posterior variance in model parameters may be drastically underestimated or overestimated. Misspecification may also lead to biased estimates. The standard assumption of independent and identically distributed (IID) Gaussian noise is applicable in some cases, but there are many other possible forms. For example, consecutive observations may be correlated due to imperfections in measurement rather than the shape of the signal itself; the magnitude of measurement noise may scale with function values; there may be time periods with higher observation volatility due to environmental variation; or even a mixture of these various types of noise within a single time series. Non-Gaussian and non-IID noise is also likely to appear in cases of time series model misspecification: when the best available model does not coincide with the hypothetical true process which generated the data, regions of poor fit may be accompanied by residual autocorrelation and spikes in the magnitude of the noise terms.

In applied circumstances, the exact noise process is never known. Some form for the noise must therefore be assumed, with consequences for inference. Whatever choice is made should have some rational basis but be flexible enough to account for the particular sample of data to hand. In this vein, parametric models likely fall short and, instead, more adaptable non-parametric methods prosper. Here, we describe a number of non- or semi-parametric models for capturing noise processes that defy characterisation into existing boxes. Through a host of toy examples with predetermined noise processes, we show that parameter inference using our noise models faithfully reproduces the true posterior distributions; that is, those distributions that result when using the correct

<sup>†</sup>In general, when  $f$  is obtained by numerically solving ODEs, some numerical error may occur.

noise process. Our noise models, and the methods for fitting them, are designed to scale well with data size and could be used across a large range of ODE models for time series. To facilitate their use, we have designed our noise models and fitting routines to work with Pints software (Clerx et al., 2019) – a general purpose Python library for fitting time series models, available from <https://github.com/pints-team/pints>. The code for this paper is available from <https://github.com/rcw5890/flexnoise>.

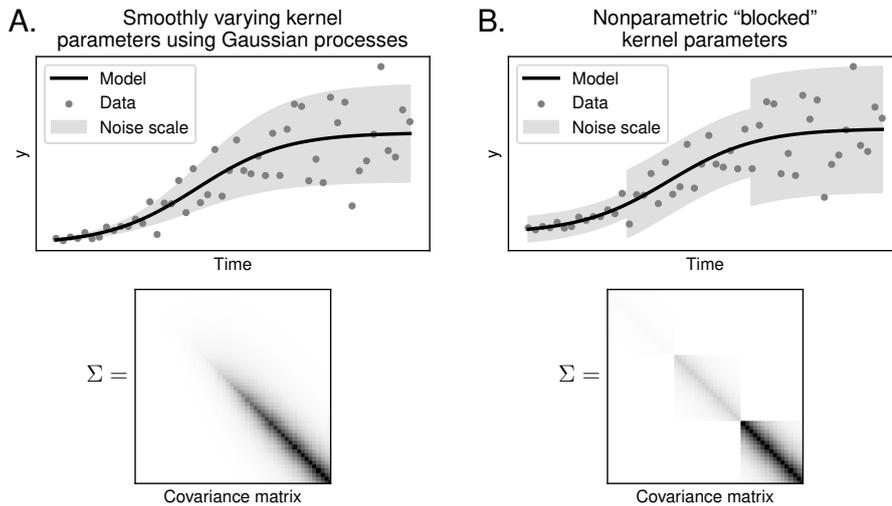


Fig. 1: **Two noise processes for time series modelling.** Panel (A) shows how non-stationary covariance kernels with continuously time-varying parameters can be used to learn the covariance matrix; and panel (B) shows how a covariance matrix can be built from non-overlapping constituent blocks.

Figure 1 gives an overview of the proposed noise model and the two different methods we use to generalise it to non-stationary noise. In both panels, time series data is illustrated with non-IID noise: the noise terms increase in magnitude as time proceeds. Panels A and B illustrate the two Bayesian methods we consider for learning appropriate covariance matrices. The first method, shown in panel A, uses a non-stationary covariance kernel whose parameters can vary continuously over time. As illustrated, this method is capable of learning a covariance matrix which steadily increases in magnitude along the diagonal. The next method, shown in panel B, divides time series into multiple regimes each with their own noise parameters. While both of these Bayesian methods allow a high degree of flexibility, they can be constrained via their prior distributions. Although we rely on a joint multivariate normal distribution for the data, and employ the Gaussian process in one of our noise processes, the methods presented here differ substantially from a Gaussian process regression, as we assume that the noise-free signal does not deviate from the given parametric model  $f$ . In §2.4, further discussion of the relationship between our methods and Gaussian processes is provided.

The remainder of this paper is organised as follows. In §2, the multivariate normal distribution is introduced as a general model for noisy time series data, and we show

how an appropriate covariance matrix can be learned from data using positive definite kernels. In §3 and §4, we describe two different methods to generalise these kernel methods to cases where the structure of noise changes over time. In §3, we describe a method where the parameters of a kernel vary smoothly over time governed by Gaussian processes. In §4, we consider clustering methods which divide a time series into different regimes, with the kernel parameters taking different values in each of these. In §5, we discuss performance considerations for long time series, and §6 shows the application of our methods to real data from experiments conducted on the hERG potassium ion channel.

## 2. The multivariate Gaussian likelihood for time series noise

The models we propose as flexible noise processes both depend on a suitably general distributional assumption governing the difference between observed and ODE-predicted data, and we use the multivariate normal. To learn the covariance matrices of the multivariate normal, we use positive definite kernel functions. This section introduces these concepts and shows how they can be used to correctly infer parameter posteriors for a time series model with stationary but non-IID noise.

### 2.1. Description of multivariate likelihood

The dataset consists of time points  $\{t_i\}_{i=1}^N$  and corresponding noisy data  $\{y_i\}_{i=1}^N$ . A typical modelling assumption is to treat the noise on each data point as IID Gaussian with a variance parameter  $\sigma^2$ , so that,

$$y_i = f(t_i; \theta) + \epsilon_i, \quad i = 1, \dots, N, \quad (2)$$

$$\epsilon_i \stackrel{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2). \quad (3)$$

Our first step is to generalize eq. (3) so that the variance of noise terms can vary (i.e. allow the noise to be non-identically distributed), and each noise realisation can be correlated with its neighbours (i.e. be non-independent). A multivariate Gaussian can handle both of these generalisations, where we model a random vector,  $\mathbf{y} = (y_1, \dots, y_N)^\top$ , as having a mean,  $\mathbf{f}(\theta) = (f(t_1; \theta), \dots, f(t_N; \theta))^\top$ ,

$$\mathbf{y} \sim \mathcal{N}(\mathbf{f}(\theta), \Sigma). \quad (4)$$

For appropriate values of the covariance matrix  $\Sigma$ , this distributional assumption encompasses a wide variety of noise forms which may include correlated and heteroscedastic noise terms. For example, eq. (4) could describe heteroscedastic noise which scales with the magnitude of the trajectory with  $\Sigma = \text{diag}(\mathbf{f}(\theta)\sigma^2)$ . For autocorrelated noise terms,  $\Sigma$  would be dense.

### 2.2. Learning the covariance matrix, $\Sigma$

Multiple methods have been proposed for inference of covariance matrices (Hoffbeck and Landgrebe, 1996; Lam and Fan, 2009; Diggle and Verbyla, 1998; Bickel and Levina, 2008; Cai and Liu, 2011; Schäfer and Strimmer, 2005). A standard Bayesian approach places

a prior on  $\Sigma$  and infers it along with ODE model parameters,  $\theta$ . Typical choices for priors include the conjugate inverse-Wishart (Gelman et al., 2013; Huang and Wand, 2013), or a prior based around the LKJ correlation matrix (Lewandowski et al., 2009; Stan Development Team, 2016). However, these methods are not designed to handle the covariance of a single time series. For a single time series obeying eq. (4), there is just one multivariate data point (that is, the vector  $\mathbf{y}$ ) available to inform the matrix  $\Sigma$ . With such limited data, standard methods for estimating covariance matrices have too much freedom, resulting in dense matrices that overfit the data.

Our strategy is to impose a positive definite covariance function  $C : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , which generates a covariance matrix according to the rule,

$$\Sigma_{ij} = C(t_i, t_j). \quad (5)$$

For example, heteroscedastic errors, where  $\Sigma = \text{diag}(\mathbf{f}(\theta)\sigma^2)$ , could be represented by the following covariance function:

$$C(t_i, t_j) = f(t_i; \theta)\sigma^2\delta_{ij}. \quad (6)$$

where  $\delta_{ij} = 1$ , if  $i = j$ ; 0, otherwise. In this paper, we consider positive definite kernels which are flexible enough to capture a wide variety of noise forms, with parameters that can, nonetheless, be learned from a single time series.

### 2.3. Kernels for time series noise

In this section, we introduce the kernels used throughout this paper. Notwithstanding the important differences discussed in §2.4, much of the work on kernel functions for Gaussian processes is applicable to ODE noise models as well, and the three kernels we discuss have seen extensive use in Gaussian process regression. One of the most widely used positive definite kernels is the Gaussian kernel (also called the Radial Basis Function, or RBF) (Fasshauer, 2011),

$$C(t_i, t_j) = \sigma^2 e^{-(t_i - t_j)^2 / 2L^2}. \quad (7)$$

We also consider the Laplacian kernel (Feragen et al., 2015) for specifying time series autocovariances, since it more faithfully reproduces the types of persistence emergent from basic univariate time series models,

$$C(t_i, t_j) = \sigma^2 e^{-|t_i - t_j| / L}. \quad (8)$$

The kernels in eqs. (7) & (8) are each characterised by two parameters which control the size and autocovariance in the errors. A more general class of kernels is the Matérn (Williams and Rasmussen, 2006),

$$C(t_i, t_j) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}}{L} |t_i - t_j| \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}}{L} |t_i - t_j| \right), \quad (9)$$

where  $K_\nu$  is the modified Bessel function of the second kind. For  $\nu = 1/2$ , the Matérn kernel simplifies to the Laplacian kernel.

#### 2.4. Comparison to Gaussian processes (GPs)

Consider a function  $g : \mathcal{X} \rightarrow \mathbb{R}$  obeying a GP with mean function  $m$  and kernel  $C$ , i.e.  $g \sim \mathcal{GP}(m, C)$  (see, for example, Rasmussen (2003)). For every finite set of inputs  $\{t_i\}_{i=1}^N$ ,  $t_i \in \mathcal{X}$ , the vector of function values  $\mathbf{g} = (g(t_1), \dots, g(t_N))^\top$  has a multivariate Gaussian distribution,

$$\mathbf{g} \sim \mathcal{N}(\mathbf{m}, \Sigma), \quad (10)$$

where  $\mathbf{m} = (m(t_1), \dots, m(t_N))^\top$  and  $\Sigma$  is generated as in eq. (5). This distribution, identical with eq. (4) for  $m(\cdot) = f(\cdot; \theta)$ , illustrates an apparent resemblance between the multivariate normal likelihood for time series noise and the GP. Our proposed noise model, however, differs from a GP in several key aspects:

- (a) In GP regression, eq. (10) determines a *prior* over functions, and the posterior over functions is inferred. Our proposed noise model uses the multivariate normal specification as a *likelihood* for finite observed data, and posterior inference applies only to the parameters of  $f$ , not the functional form of the noise-free relationship between  $y$  and  $t$  which is assumed fixed and fully determined by  $\theta$ .
- (b) To handle noisy data, Gaussian process regression typically adds an extra noise term—often IID Gaussian. No such terms are used in our multivariate normal noise process.

That is, in full, the likelihood for our multivariate normal model is given by eq. (4), with covariance matrix given by eq. (5). An example of the utility of the multivariate normal noise process is shown in Figure S1. In this example, we show that the Laplacian kernel can faithfully capture autoregressive order 1 (AR(1)) noise in an ODE time series model, enabling accurate posterior inference for the ODE model parameters.

### 3. Flexible noise for ODEs using Gaussian processes

In this section, we describe the first of two flexible noise processes which can learn effective covariance matrices from a time series. Standard positive definite kernels such as eq. (7) and eq. (8) are appropriate for simple covariance matrices. They are, however, stationary: depending only on the difference between two time points and not on absolute time. In this section, we consider models that allow kernel parameters (for example,  $\sigma$  and  $L$  in eq. (8)) to vary smoothly over time, allowing distinct sections of a time series to have different noise magnitudes and persistences. First, a brief overview of existing work on non-stationary covariance functions is provided in §3.1. In §3.2, the non-stationary version of the Laplacian kernel is presented. In §3.3, inference for non-stationary kernel parameters is introduced, and in §3.4, GP hyperparameter selection is discussed. In §3.5, results are presented on synthetic data using the non-stationary Laplacian kernel.

#### 3.1. Background on non-stationary covariance functions

Non-stationary covariance functions have been used for spatial modelling and Gaussian process regression. Unlike stationary kernels which depend only on the distance between the two inputs, in the non-stationary case, the kernel shape itself must depend on the input location. This is expressed using the notation  $k_s(u)$  for a kernel centred at location

$s$  and evaluated at location  $u$ . For example, for the Laplacian kernel with one dimensional input  $t$ , we would take

$$k_t(u) = \sigma(t)^2 e^{-|t-u|/L(t)}, \quad (11)$$

with the kernel parameters  $\sigma$  and  $L$  being functions of the kernel centre location  $t$ .

If eq. (11) is used to construct a covariance matrix, there are no guarantees that it will be positive definite. Instead, non-stationary modelling has relied on the following general formula for a non-stationary positive definite covariance function:

$$C(x_i, x_j) = \int_{\mathbb{R}^N} k_{x_i}(u) k_{x_j}(u) du, \quad (12)$$

for inputs  $x_i, x_j, u \in \mathbb{R}^N$  (Higdon et al., 1999; Paciorek, 2003). The non-stationary version of the Gaussian RBF covariance function can be derived from this formula, which has been used in non-stationary Gaussian process regression (Gibbs, 1998; Paciorek and Schervish, 2004). To learn time-varying kernel parameters, a Gaussian process prior can be placed on each (Paciorek and Schervish, 2004; Heinonen et al., 2016).

### 3.2. Non-stationary Laplacian covariance function

In this section, we present a non-stationary version of the Laplacian kernel. The techniques presented here are equally applicable to any appropriate positive definite kernel, however.

The one-dimensional non-stationary Laplacian covariance function is:

$$C(t_i, t_j) = \sigma(t_i)\sigma(t_j) \sqrt{\frac{2L(t_i)L(t_j)}{L(t_i)^2 + L(t_j)^2}} \exp\left(-\frac{|t_i - t_j|}{\sqrt{L(t_i)^2 + L(t_j)^2}}\right). \quad (13)$$

Eq. (13) may be derived as a special case of the non-stationary Matérn kernel (Paciorek and Schervish, 2004); it also follows directly from the one-dimensional case of eq. (12) using reparametrised versions of the respective stationary kernels, with the reparametrisations chosen to ensure that the final non-stationary covariance functions have a sensible form (cf. eq. (3.69) in Gibbs (1998)). The logarithms of  $L$  and  $\sigma$  each vary over time governed by Gaussian process priors:

$$\log L \sim \mathcal{GP}(\mu_l, K_l), \quad \log \sigma \sim \mathcal{GP}(\mu_\sigma, K_\sigma), \quad (14)$$

where  $\mu_l$ ,  $K_l$ ,  $\mu_\sigma$ , and  $K_\sigma$  are the GP hyperparameters.

### 3.3. Inference

Having specified a non-stationary covariance function such as eq. (13), the next task is to infer the posterior distribution of model and covariance parameters. However, analytic expressions for the posterior mean and variance of the Gaussian processes  $L(t)$  and  $\sigma(t)$  are not available. Instead, MCMC sampling or maximum a posteriori (MAP) estimation can be used to infer the values of  $L(t)$  and  $\sigma(t)$  at each time point (Heinonen et al., 2016). MCMC sampling yields a set of samples distributed according to the posterior distribution, while MAP estimation uses optimisation to find the parameters

values at which the posterior distribution is maximised. For both MCMC and MAP estimation, we recommend the use of gradient-based methods (e.g., Hamiltonian MCMC and gradient-based optimisers) for improved convergence rates in the high dimensional parameter space (Neal, 2011). When analytic gradients are not available, automatic differentiation can be used. Indeed, all our GP examples presented here involve an interpolation scheme discussed in §5, rendering analytic derivatives intractable, and we resort to using automatic differentiation.

MCMC sampling is advantageous because it can, in theory, fully characterise posterior uncertainty in the covariance parameter fits. However, it is computationally costly and may not be necessary for concentrated unimodal posteriors. In these cases, MAP estimation may be preferred due to its lower computational burden. While MAP estimates are often sufficient for the kernel parameter fits, we typically still require information about the posterior uncertainty in the ODE model parameters. Thus, we recommend the following procedure for long ODE time series problems with non-stationary covariance functions, which is specified in Algorithm 1. First, the joint MAP estimate of model parameters and covariance parameters is obtained using a gradient-based optimiser. Then, MCMC sampling is used to obtain the posterior distribution of model parameters conditional on the previously obtained MAP estimate of covariance parameters. For both optimisation and MCMC sampling, random or uniform initialisation of the covariance parameters will work for easier problems but will delay convergence on longer time series. In long time series problems with intelligible noise patterns, we recommend a data-driven initialisation of the covariance parameters in order to accelerate convergence of MCMC or optimisation. To initialise  $L$  and  $\sigma$  in the non-stationary Laplacian covariance function, we use the procedure given by Algorithm 2. In practice, gradient-based optimisers such as L-BFGS-B (Zhu et al., 1997) may settle at local maxima. Thus, we perform optimisation with multiple restarts, with each restart taking a different initial value. A set of variable yet plausible initial values for the restarts can be generated by rerunning

Algorithm 2 multiple times with different sliding window widths.

---

**Algorithm 1:** MCMC estimates of ODE parameters using MAP estimates for kernel parameters.

---

**Input:** A parametric model  $f(t; \theta)$ , a non-stationary covariance function  $C_\phi(t_i, t_j)$ , and observed data  $\{y_i\}_{i=1}^N$  at time points  $\{t_i\}_{i=1}^N$ ;  $\phi$  indicates the full set of kernel parameters defining  $C$ .

**Output:** MCMC samples distributed according to the conditional posterior  $p(\theta|y, \phi_{\text{MAP}})$ .

Initialise  $\phi$ , for example using to Algorithm 2 for the Laplacian kernel;

Use gradient-based optimisation to find  $(\theta_{\text{MAP}}, \phi_{\text{MAP}}) = \arg \max p(\theta, \phi|y)$ ;

Calculate the fixed covariance matrix  $\Sigma_{\text{MAP}}$  such that  $\Sigma_{i,j} = C_{\phi_{\text{MAP}}}(t_i, t_j)$ ;

Use the covariance matrix defined above to form the likelihood

$\mathcal{N}(y|f(t; \theta), \Sigma_{\text{MAP}})$ ;

Use MCMC to sample from the conditional posterior  $p(\theta|y, \phi_{\text{MAP}})$

---

**Algorithm 2:** Initialisation for non-stationary Laplacian kernel parameters.

---

**Input:** A parametric model  $f(t; \theta)$  and observed data  $\{y_i\}_{i=1}^N$  at time points  $\{t_i\}_{i=1}^N$  with spacing  $\Delta t$ . A sliding window width for each kernel parameter.

**Output:** A rough estimate of the parameters  $\{\sigma_i\}_{i=1}^N$  and  $\{L_i\}_{i=1}^N$  of the non-stationary Laplacian kernel which can be used to initialise an optimisation algorithm or MCMC sampler.

Use optimisation to find the MAP estimate of model parameters assuming an IID noise model,  $\theta_{\text{MAP, IID}}$ ;

Subtract  $f(t; \theta_{\text{MAP, IID}})$  from the observed data to obtain an estimate of the noise terms  $\epsilon_i$ ;

At each time point  $t_i$ , calculate the empirical variance  $v_i$  and 1st order autocorrelation  $\rho_i$  of the noise terms within a sliding window centred on that time point;

Smooth both estimates using a Wiener filter (Wiener, 1950);

At each time point,  $t_i$ , set  $\sigma_i = \sqrt{v_i}$  and  $L_i = -\Delta t / \log(|\rho_i|)$

---

### 3.4. Gaussian process hyperparameters

For the GPs defined by eqs. (14), we used squared exponential kernels with constant mean functions (Heinonen et al., 2016). With this assumption, there are six Gaussian process hyperparameters for the model (for each of  $L$  and  $\sigma$ , a mean  $\mu$ , noise level  $\alpha$ , and length scale  $\beta$ ). Prior knowledge or a grid search can be used to set these values, although existing work suggests only  $\beta$  has a substantial effect on posteriors in most cases (Heinonen et al., 2016). We set  $\mu = 0$  and  $\alpha = 1$  (Heinonen et al., 2016), and propose the following method to set  $\beta$ , which controls how the Gaussian process can change over time. This behaviour is crucial to the adaptivity of the method. If the scale,  $\beta$ , is too short, the GP will overfit local fluctuations; too large and it will fail to account for real changes in the process over time.

To set the length scale, we used a heuristic based on the expected rate of change of

the noise process. Given evenly spaced time points with spacing  $\Delta t$  and a user-specified number of time points  $N_c$ , we set  $\beta$  as the solution of

$$\zeta = e^{-(N_c \Delta t)^2 / (2\beta^2)}, \quad (15)$$

for some small value  $\zeta = 0.01$ . This equation imposes that the prior covariance between two values of the Gaussian process  $N_c$  time points apart is close to zero, thus summarising the prior belief that the noise structure can change over that time scale. Good choices for  $N_c$  will generally be problem specific. For non-uniform spacing,  $N_c \Delta t$  could be replaced by an appropriate time interval.

### 3.5. Example with synthetic data

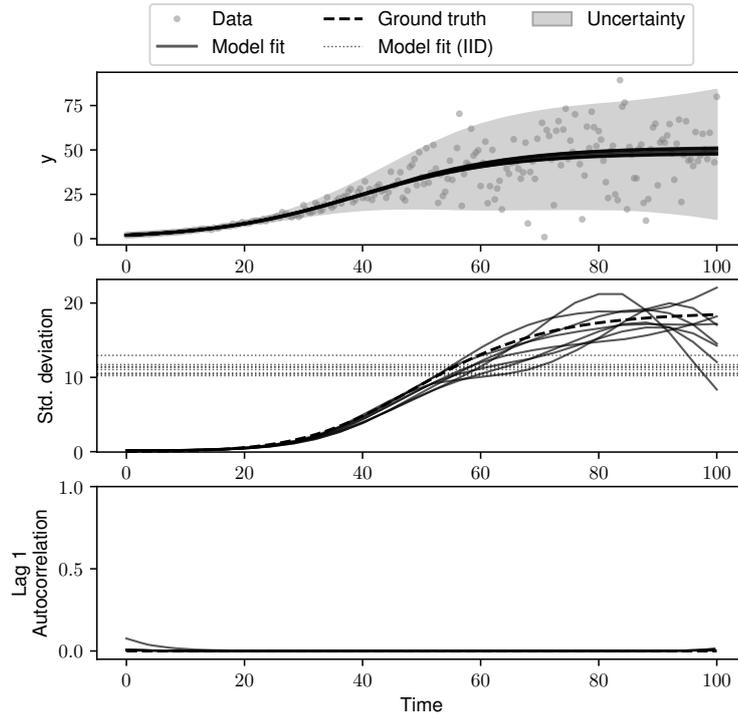
In this example, we use the two-parameter logistic growth model:

$$\frac{dy}{dt} = ry(1 - y/K). \quad (16)$$

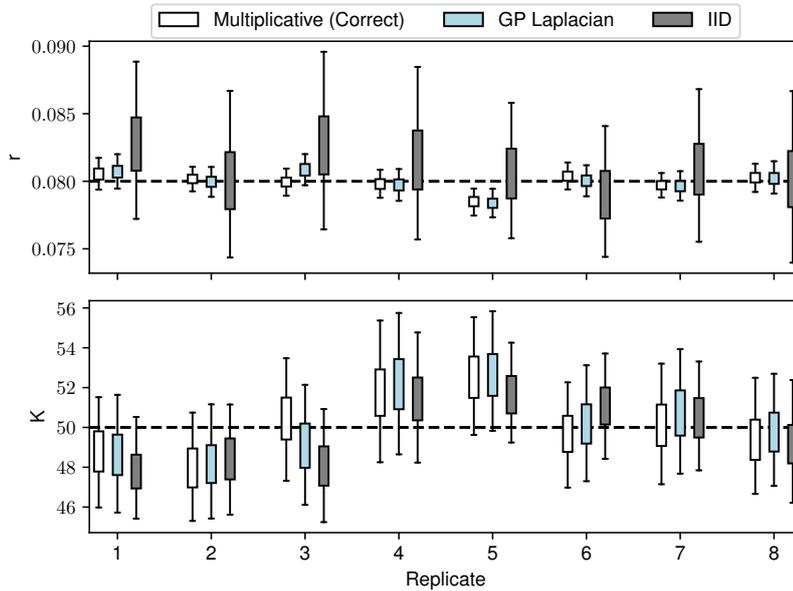
We demonstrate the results of fitting the non-stationary kernel to synthetic data, generated from a logistic growth model with  $r = 0.08$ ,  $K = 50$ , and  $f(t = 0) = 2$  with multiplicative Gaussian noise:

$$y_i = f(t_i; \theta) + f(t_i; \theta)^\eta v_i, \quad (17)$$

where  $y_i$  is an observed data point,  $f(t; \theta)$  is the ODE model solution, and  $v_i \stackrel{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2)$ . We set  $\eta = 2$  and  $\sigma = 0.0075$ . Eqs. (16)&(17) were used to generate eight replicate time series, each with 250 time points. We considered parameter inference for each set of series under three different noise processes: multiplicative (i.e. the true noise process), the non-stationary Laplacian kernel, and IID Gaussian. In each case, Algorithm 1 was used to generate posterior samples from  $r$  and  $K$ . MCMC sampling for model parameters was performed using Pints inference software (Clerx et al., 2019) with three Markov chains and a total of 20,000 iterations on each. The posterior was sufficiently simple here that a non-gradient-based sampler – a form of adaptive covariance algorithm, called the Haario Bardenet method (Haario et al., 2001; Johnstone et al., 2016) – was used opposed to Hamiltonian Monte Carlo. On a desktop processor, each chain took approximately 20 minutes to run. The first half of each chain was discarded as warm-up, and convergence was assessed using the Gelman  $\hat{R}$  statistic, requiring  $\hat{R} < 1.05$  for all parameters (Gelman et al., 2013). To set the GP hyperparameter  $\beta$ , we used eq. (15) with  $N_c = 200$ . The results are shown in Figure 2. In panel (a), the data (from the first replicate) is shown in the top panel, along with the fitted model trajectory. Below, the standard deviation and lag 1 autocorrelation are shown based on the MAP estimates for each replicate and indicate good correspondence with the ground truth. In panel (b), the posterior distributions for the model parameters are shown. The growth parameter,  $r$ , was most affected by incorrectly assuming IID Gaussian noise, where the IID noise model resulted in estimates with overly inflated uncertainty. This is because model output is most sensitive to  $r$  in the first half of the series, where the IID noise model overestimates the noise level. In all cases, the GP method provided a higher fidelity estimate of uncertainty than IID noise; in most cases the location of the posterior is also improved. Another



(a)



(b)

Fig. 2: **Non-stationary Laplacian kernel fits to logistic data.** The top plot of panel (a) shows an example logistic growth time series with multiplicative noise, with 250 time points. In the other two plots of (a) and in panel (b), results for model fits to eight replicate datasets are shown. In the middle plot of panel (a), the true standard deviation  $\sqrt{C(t_i, t_i)}$  is shown, along with model estimates of it at the MAP estimates for  $L$  and  $\sigma$  (one line per each replicate). In this plot, we also indicate the standard deviations estimated by the IID assumption as horizontal dashed lines. In the bottom plot of panel (a), the same is shown as in the middle plot, except with results for the lag 1 autocorrelation,  $C(t_i, t_{i+1})/(\sigma(t_i)\sigma(t_{i+1}))$ . Panel (b) shows MCMC estimates of the posterior distributions for the logistic growth model parameters under three different assumptions for the noise process; the boxes cover the central 50% posterior estimates, while the whiskers cover the central 95% posterior estimates, and the dashed lines indicate the true values of the parameters.

example of the GPs fitted to synthetic data is given in Figure S2. In this example, the true data generating process consists of discrete blocks of different noise models, and the results show the ability of the non-stationary kernel method to find an appropriate smooth approximation.

#### 4. Flexible noise for ODEs using change points

In this section, we describe a second flexible noise process for learning covariance matrices for time series noise. In §4.1, an overview of our “block covariance matrix” method is provided along with a change point model. In §4.2, we use synthetic data from the logistic model to illustrate how the model can be fitted.

##### 4.1. Nonparametric change point models

Instead of assuming that kernel parameters vary smoothly and continuously over time, we now introduce a model that divides the time domain into discrete sections; each with a distinct noise model. A model of this type results in a block structure for the covariance matrix. Assuming the time series is divided into  $M$  regimes indexed by  $m = 1, \dots, M$ , and each regime  $m$  has a covariance kernel  $k_m$ , the covariance matrix takes the form,

$$\Sigma = \begin{pmatrix} \Sigma^{(1)} & 0 & \dots & 0 \\ 0 & \Sigma^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Sigma^{(M)} \end{pmatrix}, \quad (18)$$

where each block  $\Sigma^{(m)}$  is a positive definite matrix, with elements given by  $\Sigma_{ij}^{(m)} = k_m(t_i, t_j)$ . The covariance matrix from eq. (18) is guaranteed to be positive definite because each block is positive definite. Eq. (18) requires that the autocorrelation between consecutive points at the boundary between two blocks is zero. When autocorrelation varies more gradually, this may be a disadvantage, and could likely be better handled by the non-stationary kernel method in §3. But, for more rapid changes in covariance, the block method should outperform.

To infer the locations of block boundaries is a type of “change point” problem (see, for example, Aminikhanghahi and Cook (2017)), which is a wide field with many methods and fitting processes. We choose a nonparametric approach to change point detection, which has the benefit that the number of boundaries need not be fixed beforehand. In particular, we specify the “restricted product partition model” (PPM) induced by the Pitman-Yor process as a prior on the partitions (Martínez and Mena, 2014). The PPM approach is convenient for learning model parameters, as the posterior over partitions can be inferred jointly with the posterior over ODE model parameters.

To specify the change point model, we place a prior over the partitions which lie in the class of set compositions of the index set  $[N] = \{1, 2, \dots, N\}$ , i.e.,  $\mathcal{C}_{[N]}$ . This class contains all partitions  $\{S_1, S_2, \dots, S_M\}$  such that  $S_m = \{s_j + i : i = 1, 2, \dots, n_m\}$ , where  $s_0 = 0$ ,  $s_j = n_1 + n_2 + \dots + n_j$ , and  $n_m$  is the number of indices in  $S_m$  (Martínez and Mena, 2014). As an illustrative example, consider the case where  $N = 3$ , in which  $\mathcal{C}_{[3]}$

contains the following elements:

$$\{\{1\}, \{2\}, \{3\}\}, \{\{1, 2\}, \{3\}\}, \{\{1\}, \{2, 3\}\}, \{\{1, 2, 3\}\}.$$

$\mathcal{C}_{[N]}$  contains all admissible assignments of the time points to consecutive blocks.

Let  $\rho_N$  indicate a random variable taking values in  $\mathcal{C}_{[N]}$ . For a problem with  $N$  time points and  $k$  regimes, the PPM prior on  $\rho_N$  with discount parameter  $s$  and strength parameter  $\phi$  is given by:

$$p(\rho_N | s, \phi) = \left( \frac{N! \prod_{i=1}^{k-1} (\phi + is)}{k! (\phi + 1)_{N-1\uparrow}} \prod_{j=1}^k \frac{(1-s)_{n_j-1\uparrow}}{n_j!} \right). \quad (19)$$

In this equation,  $x_{n\uparrow}$  denotes the rising factorial or Pochhammer function; i.e.,  $x_{n\uparrow} = (x)(x+1)\dots(x+n-1)$ . Some visual analysis of the prior  $p(\rho_n)$  is presented in Figure S3;  $\phi$  serves as a location parameter controlling the number of blocks, while  $s$  is a concentration parameter controlling the variance in this number.

We assume a parametric form for all block kernels  $k_m$ , such as the Laplacian kernel given in eq. (8) (which takes two parameters). These parameters are constant within a block, but vary from block to block. The appropriate priors for the kernel parameters will depend on the form of the kernel. For the Laplacian kernel, we place diffuse normal priors on the logarithms of the parameters:

$$\log L \sim \mathcal{N}(-4, 4), \log \sigma \sim \mathcal{N}(0, 4). \quad (20)$$

MCMC inference for the posterior is possible using Metropolis-Hastings steps in a split-merge-shuffle sampler, as used in Algorithm 2 of Martínez and Mena (2014). Each iteration of this approach consists of a series of proposals that may be accepted or rejected: first, there is a proposal that affects the number of blocks; this can be either a merge of two consecutive blocks or a split at some random point within an existing block; second, a proposal that keeps the number of blocks the same is used – in this “shuffle” proposal, the boundary between two blocks is randomly moved somewhere within their limits.

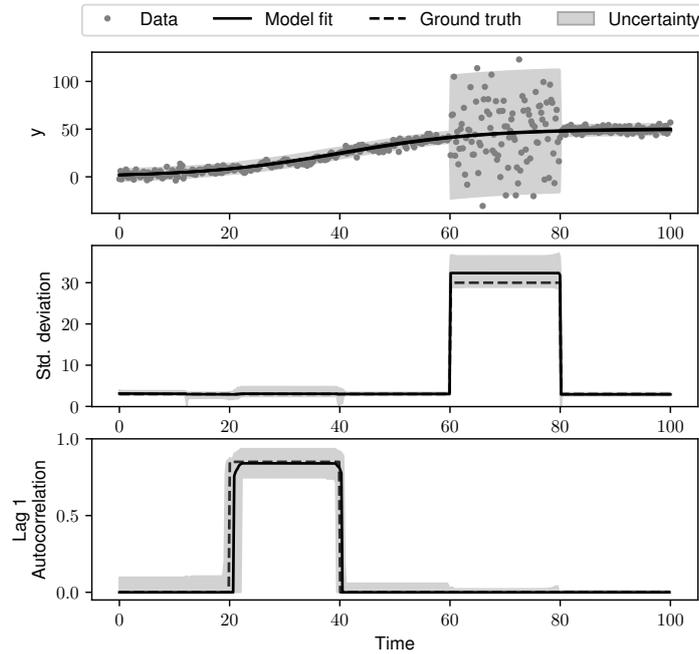
In prior work on inference for PPM models (Martínez and Mena, 2014), the MCMC algorithm is restricted to the case where all within-block parameters can be integrated out, allowing calculation of the marginal posterior probability  $p(\rho_N | y) \propto p(y | \rho_N) p(\rho_N)$  for any valid proposed partition  $\rho_N$ . In this work, we relax this assumption, so that we can only calculate the likelihood given explicit values for both the partition and the block parameters. This presents a challenge to inference, as evaluations of the marginal likelihood are essential to calculate the acceptance ratio of the merge and split proposals. In this work, we modify the split-merge-shuffle sampler to handle the non-conjugate case using the ideas of the Sequentially-Allocated Merge-Split (SAMS) Sampler for Dirichlet processes (Dahl, 2005). Briefly, when proposing a split, we simultaneously propose new values for the block parameters in the newly split block using a random walk. When proposing a merge, the current parameter values from the first of the two merged blocks are selected for use in the proposal. At each MCMC iteration, an additional Metropolis-Hastings step is performed to update the explicit block parameter values. The details of our MCMC algorithm are given in §S4.

A key benefit of a nonparametric approach for covariance matrix estimation is that it can learn an appropriate number of blocks from data. This flexibility has a cost, however – in theory, each individual time point could be assigned to its own block. To avoid this outcome, informative priors on  $s$  and  $\phi$  can be specified. For  $s$ , we specify the prior:  $s \sim \text{beta}(1, 1)$ . For  $\phi$ , we place a shifted gamma prior<sup>‡</sup> with parameters  $(a, b, s)$ . We choose the hyperparameters  $a, b$  such that, at the prior mean of  $s$ , prior mass in the number of blocks is heavily concentrated at one block. Appropriate values for  $a$  and  $b$  may thus depend on the length of the time series and the desired strength of the prior preference for one block; we achieve a moderately informative prior using  $a = 0.01$  and  $b = 100$ . This specification represents a weak null belief that the noise process is stationary, but allows the noise process flexibility to change if necessary.

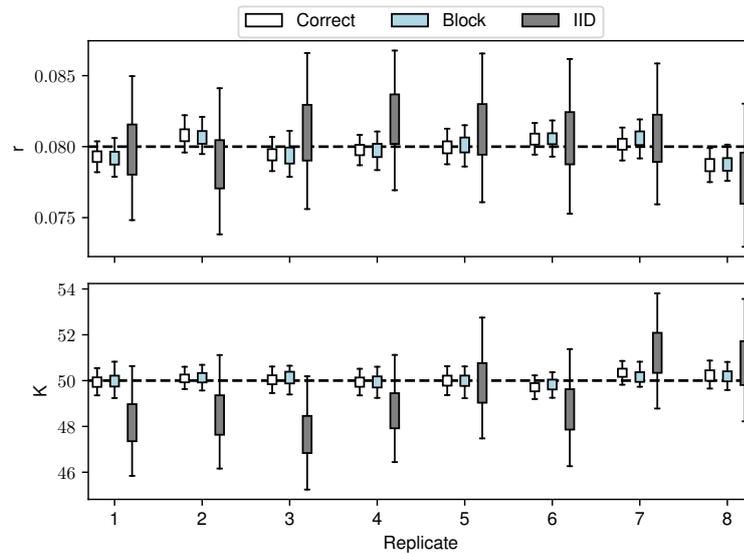
#### 4.2. Example with synthetic data

We tested the block noise process on synthetic data generated according to the logistic model, eq. (16), with  $r = 0.08$ ,  $K = 50$ , and  $f(t = 0) = 2$ . In addition, we contrived a noise process with 5 regimes, each of 100 consecutive time points. The first, third and fifth regimes had IID Gaussian noise with  $\sigma = 3$ , the second regime had AR(1) noise with  $\rho = 0.85$  and  $\sigma = 3$ , and the fourth regime had IID Gaussian noise with  $\sigma = 30$ . A total of 20,000 MCMC iterations were performed using our version of the split-merge-shuffle sampler (see §S4), with the first 10,000 discarded as warm up, and convergence was assessed using the Gelman  $\hat{R}$  statistic, requiring  $\hat{R} < 1.05$  for all model parameters. Each MCMC iteration consisted of one split-merge-shuffle step for the blocks as well as one adaptive covariance step for the model parameters. On this dataset, all iterations took approximately 100 minutes to run on a desktop processor. In the first panel of Figure 3 (a), we show the data and the posterior median of the learned model trajectory, as well as the posterior median of two standard deviations of the inferred noise process about the model trajectory. The next two panels compare the estimated error standard deviation and lag 1 autocorrelation with their true values. This shows that the change point flexible noise process readily captures the five different regimes and learns their boundaries. In Figure 3 (b), we plot the posterior distributions for the logistic growth parameters with three different assumptions for the noise process. The noise process labelled “Correct” indicates an assumption of a multivariate normal likelihood with a block covariance matrix as given in eq. (18), with the block locations and sizes fixed to their correct values. However, in the “Correct” comparator method, the kernel parameters within each fixed block are not known, and are inferred jointly with the ODE model parameters. Meanwhile, the results from the nonparametric block covariance method are labelled with “Block”. For both  $r$  and  $K$ , the IID noise model results in inflated estimates of posterior variance, while both the block covariance method and the true noise model recover precise posteriors.

<sup>‡</sup>If  $X - c \sim \text{gamma}(a, b)$ , then  $X \sim \text{shifted-gamma}(a, b, c)$ .



(a)



(b)

Fig. 3: **Block covariance kernel fits to logistic data.** Panel (a) shows an example logistic growth time series with 5 blocks of different noise forms and a total of 500 time points. The model fit  $\pm 2$  standard deviations of the inferred noise process are overlaid on the data points. In the middle and bottom plots of panel (a), the true values of standard deviation and lag 1 autocorrelation are shown as dotted lines, while the inferred posterior median standard deviation and central 90% posterior are shown as a solid line and shaded region. In panel (b), results for model fits to eight replicate datasets are shown. This panel shows MCMC estimates of the posterior distributions for the logistic growth model parameters under three different assumptions for the noise process; the boxes cover the central 50% posterior estimates, while the whiskers cover the central 95% posterior estimates, and the dashed lines indicate the true values of the parameters.

## 5. Efficient computation for long time series

In real-life time series problems, we often encounter numbers of data points in the hundreds or thousands. In these cases, the computational cost of the methods mentioned above becomes a serious hindrance. In this section, we thus provide two computational strategies which can significantly decrease the runtime, and enable scaling to long time series.

The computational cost of the multivariate normal likelihood given in eq. (4) is sensitive to the number of time series points,  $n$ : the covariance matrix,  $\Sigma$ , is  $n \times n$ , and the multivariate normal requires its inverse and determinant to be calculated. For highly sampled time series data,  $n$  is large enough that these computations are impossible.

To scale to long time series, we take advantage of the relative sparsity of the covariance matrix. Any reasonable kernel, including the kernels we study in this paper, will generate matrices whose values are close to zero sufficiently far away from the diagonal. We truncate the entries in our covariance matrix, setting all those below a small threshold ( $10^{-9}$ ) to zero. This results in a sparse matrix whose inverse and determinant can be computed using sparse Cholesky decomposition.

The non-stationary kernel for the GP method presents another scaling challenge as it requires inferring the value of the various GPs at each time point. For the non-stationary Laplacian kernel, this means that  $L(t)$  and  $\sigma(t)$  in eq. (14) are estimated for all  $t$ . For long time series, the number of parameters to infer then becomes prohibitive. To reduce this cost, we infer only the GP posterior on a sparser grid of time points. The GP functions are then interpolated to populate the covariance matrix at the original time points; here, we use linear interpolation but recognise that, if the GP value changes rapidly, more nuanced schemes may be appropriate. By introducing the interpolation step, the analytical calculation of the gradient of the multivariate normal likelihood becomes intractable. Therefore, in order to use gradient-based optimisers or MCMC samplers (such as L-BFGS-B and Hamiltonian Monte Carlo) (Zhu et al., 1997; Neal, 2011), we rely on automatic differentiation.

The specific speedup enabled by these two computational approximations will vary greatly according to details of the problem at hand but, in our experience, can be quite dramatic. With a time series of length 150, we found that learning the non-stationary Laplacian kernel parameters at every fifth point and then interpolating resulted in a speedup of approximately 500% at each MCMC iteration, and using sparse covariance matrices resulted in a speedup of approximately 4100% for evaluation of the multivariate normal likelihood. On a typical desktop computer, these approximations enable reasonable runtimes for time series with lengths on the order of 10,000 points, as we demonstrate in the following section.

## 6. Application to hERG channel kinetics

The preceding examples of the flexible noise processes used synthetic data; in this section, we fit flexible noise processes to real data generated from experiments on the hERG potassium ion channel. This problem is challenging because the noise is clearly not IID, and, also because there may be misspecification of the underlying ODE model. In §6.1, we provide a brief description of the hERG channel and a model used to investigate its

behaviour and also describe experimental data generated for this system. In §6.2, we show how a flexible noise process can capture non-IID noise trends leading to different estimates of model parameters compared to those from an IID noise model.

The hERG channel time series are long (7700 time points, after  $10\times$  thinning), and we expect that the variation in the magnitude and autocorrelation of their noise terms can be captured using a continuously varying method. Thus, in this section we use the non-stationary covariance kernel method from §3 along with those modifications given in §5 to allow efficient computation.

### 6.1. Description of hERG problem

The *human Ether-à-go-go-Related Gene* (hERG) encodes the alpha subunit of the potassium channel Kv11.1 that conducts the rapid delayed rectifier potassium current  $I_{Kr}$ . This current is of great importance in cardiac electrophysiology and safety pharmacology; reduction of  $I_{Kr}$  by pharmaceutical compounds or mutations can induce fatal disturbances in cardiac rhythm. Interest in this model generally centres on understanding the current response of the hERG channel when a voltage stimuli  $V$  is applied. The current can be described with a Hodgkin & Huxley-style structure model (Hodgkin and Huxley, 1952) given by:

$$I_{Kr} = g_{Kr} \cdot a \cdot r \cdot (V - E_K), \quad (21)$$

where  $g_{Kr}$  is the maximal conductance, and  $E_K$  is the reversal potential (Nernst potential) for potassium ions which can be calculated directly from potassium concentrations using the Nernst equation.

The kinetic terms of the model,  $a$  and  $r$ , are governed by:

$$\frac{da}{dt} = \frac{a_\infty - a}{\tau_a}, \quad \frac{dr}{dt} = \frac{r_\infty - r}{\tau_r}, \quad (22)$$

$$a_\infty = \frac{k_1}{k_1 + k_2}, \quad r_\infty = \frac{k_4}{k_3 + k_4}, \quad (23)$$

$$\tau_a = \frac{1}{k_1 + k_2}, \quad \tau_r = \frac{1}{k_3 + k_4}, \quad (24)$$

where,

$$k_1 = p_1 \exp(p_2 V), \quad k_3 = p_5 \exp(p_6 V), \quad (25)$$

$$k_2 = p_3 \exp(-p_4 V), \quad k_4 = p_7 \exp(-p_8 V). \quad (26)$$

The model has 9 parameters  $\theta = (g_{Kr}, p_1, p_2, \dots, p_8)$  to be inferred, all of which are positive. These parameters are the maximal conductance  $g_{Kr}$  [pS] and kinetic parameters  $p_1, p_2, p_3, \dots, p_8$  [ $s^{-1}$ ,  $V^{-1}$ ,  $s^{-1}$ ,  $\dots$ ,  $V^{-1}$ ].

Experimental data of the current are taken from a freely available dataset (Lei et al., 2019a; 2019b), where the voltage stimuli  $V$  were designed for parametrising the model.

The logarithm-transformation was applied to all model parameters  $\theta$ , such that the transformed parameters  $\phi = \log(\theta)$  are unconstrained. To account for the impact of this non-linear transformation on the posterior, a Jacobian transformation was applied.

Priors for  $\phi$  were selected using existing literature results (Lei et al., 2019a; 2019b), and, for each element of  $\phi$ , a weakly informative prior Gaussian distribution was used (see Table S1 for the prior hyperparameters).

## 6.2. Results

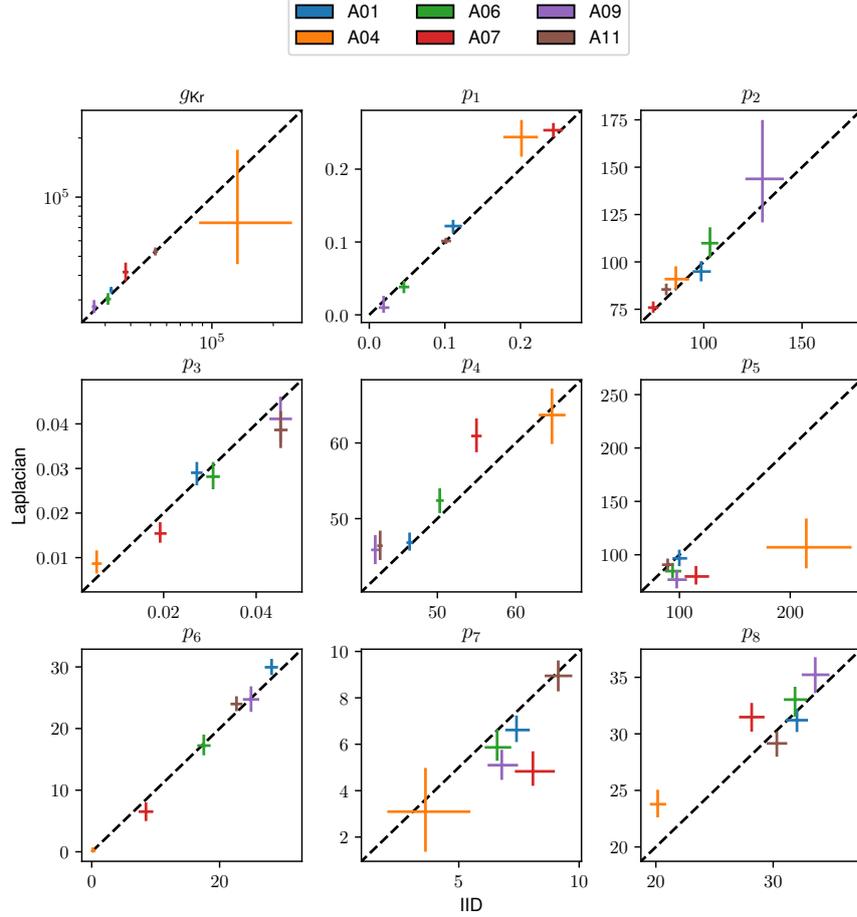


Fig. 4: **Posterior distributions for hERG model parameters.** This figure compares the posterior distributions resultant from the IID Gaussian noise assumption (“IID”) and non-stationary Laplacian kernel (“Laplacian”) for the nine hERG model parameters for six cells. For each parameter, the central 95% range of the posterior is shown for each noise model as a bar, with the IID posterior shown on the horizontal axis and the non-stationary posterior shown on the vertical axis. Within each plot, a diagonal dashed line is drawn along  $y = x$ .

For six different cells, the model parameter posteriors were obtained via MCMC

using the IID noise model and the non-stationary Laplacian kernel flexible noise model. To obtain posterior samples, the simulated tempering population MCMC algorithm was used (Jasra et al., 2007), with convergence assessed using the Gelman  $\hat{R}$  statistic (Gelman et al., 2013), and the first half of each chain discarded as warm up. For the non-stationary Laplacian kernel, we used Algorithm 1 for inference and Algorithm 2 for initialisation. The fits for each of six cells are shown with the time series data in Figure S4.

Figure 4 shows the central 95% posterior distribution ranges for all nine model parameters, assuming either IID Gaussian noise (horizontal axis) or the non-stationary noise process (vertical axis). There were significant differences in the parameter estimates for almost all parameters, with much of probability mass not overlapping the IID=Laplacian line. Additionally, the more sophisticated noise model resulted in substantially higher posterior variance for several model parameters, notably including  $g_{Kr}$ ,  $p_2$  and  $p_4$ . Cell A04 is an outlier: this is likely because this cell has a region of drastic misspecification in much of the time series, from  $t = 6$  to  $t = 10$ . While the model fits for all six cells indicate short regions of misspecification, which is particularly apparent after the drops in current around  $t = 2$  and  $t = 14$ , cell A04 (and to a lesser extent, A07) suffer from more extensive misspecification. The data and inferred fits for cell A04 are shown in Figure 5. The non-stationary noise model detects the central misspecified region by assigning high variance and autocorrelation in the middle of the time series. In the time series for cells A04 and A07, the poor fit between model and data may be largely explained by the fact that our model in this study (§6.1) fails to account for experimental artefacts in the voltage clamp experiment, such as leakage current—these artefacts may explain much of the cell-to-cell variability observed in these experiments (Lei et al., 2020). Thus, the high levels of standard deviation and autocorrelation detected in these time series suggest that a more detailed model of the experiment is necessary in order to understand these cells and correct the regions of obvious poor fit.

## 7. Discussion

When performing Bayesian inference for the parameters of time series models, the assumption made for the noise process may drastically alter the posterior estimates of parameter uncertainty. The flexible noise models described in this paper have the ability to learn noise processes from the data, including complex, non-stationary noise processes. The utility of these methods has been demonstrated in constructed synthetic data examples.

In applied circumstances, noise terms which exhibit autocorrelation and time-varying magnitude often indicate model misspecification. This is what we observe in the hERG time series problem, in which the best fit model trajectory cannot fully express the signal that is clear in the data. In these cases, our non-stationary covariance noise process is able to pick out the regions of poor fit and model the spikes in magnitude and autocorrelation present at those time periods, with corresponding changes apparent in the model parameter posteriors. Future work to better handle misspecification in time series problems such as the hERG channel may benefit from the ability, offered by the methods in this paper, to avoid the often incorrect assumption of IID Gaussian noise.

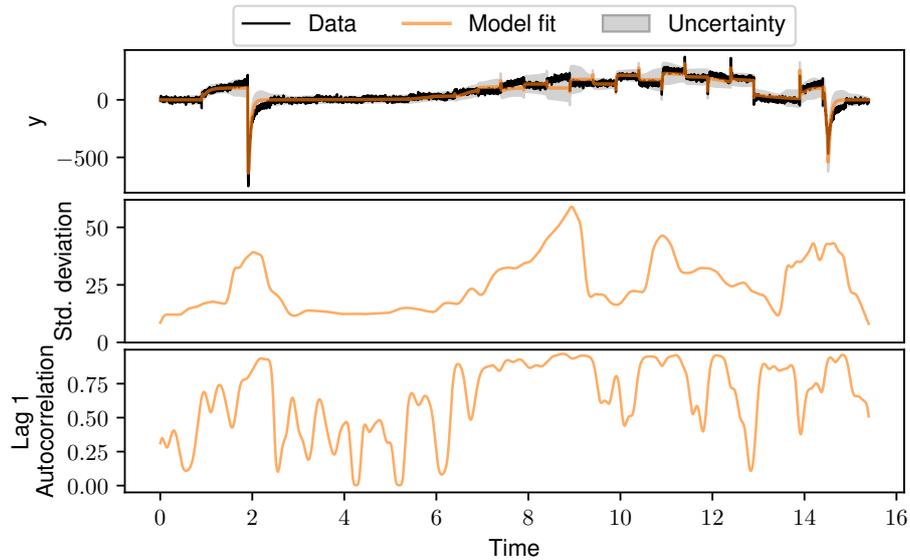


Fig. 5: **Non-stationary Laplacian kernel noise model fit to hERG cell A04.** This figure shows the data and model fit (top panel), and the MAP estimate standard deviation and lag 1 autocorrelation over time (second and third panels) inferred by the non-stationary Laplacian kernel noise model for cell A04.

## 8. Author contributions

RC, BL, CLL, MR, and DG conceived the methods. RC designed the simulations and developed the algorithms and code. CLL set up the hERG data and model simulation code. RC, BL, and CLL wrote the manuscript. BL, MR, and DG supervised the analysis. All authors reviewed and approved the manuscript.

## 9. Acknowledgements

RC acknowledges support from the EPSRC. CLL acknowledges support from the Clarendon Scholarship Fund, and the EPSRC, MRC [EP/L016044/1] and F. Hoffmann-La Roche Ltd. for studentship support.

## References

- Aminikhanghahi, S. and D. J. Cook (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems* 51(2), 339–367.
- Bickel, P. J. and E. Levina (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics* 36(1), 199–227.
- Cai, T. and W. Liu (2011). Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association* 106(494), 672–684.

- Clerx, M., M. Robinson, B. Lambert, C. Lei, S. Ghosh, G. Mirams, and D. Gavaghan (2019). *Journal of Open Research Software* 7(1).
- Dahl, D. B. (2005). Sequentially-allocated merge-split sampler for conjugate and non-conjugate dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 11(6).
- Diggle, P. J. and A. P. Verbyla (1998). Nonparametric estimation of covariance structure in longitudinal data. *Biometrics*, 401–415.
- Fasshauer, G. E. (2011). Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation* 4, 21–63.
- Feragen, A., F. Lauze, and S. Hauberg (2015). Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3032–3042.
- Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013). *Bayesian data analysis*. CRC press.
- Gibbs, M. N. (1998). *Bayesian Gaussian processes for regression and classification*. Ph. D. thesis, University of Cambridge.
- Haario, H., E. Saksman, J. Tamminen, et al. (2001). An adaptive Metropolis algorithm. *Bernoulli* 7(2), 223–242.
- Heinonen, M., H. Mannerström, J. Rousu, S. Kaski, and H. Lähdesmäki (2016). Non-stationary Gaussian Process Regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pp. 732–740.
- Higdon, D., J. Swall, and J. Kern (1999). Non-stationary spatial modeling. In *Proceedings of the Sixth Valencia International Meeting on Bayesian Statistics*, pp. 761–768.
- Hodgkin, A. L. and A. F. Huxley (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology* 117(4), 500–544.
- Hoffbeck, J. P. and D. A. Landgrebe (1996). Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(7), 763–767.
- Huang, A. and M. P. Wand (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis* 8(2), 439–452.
- Jasra, A., D. A. Stephens, and C. C. Holmes (2007). On population-based simulation for static inference. *Statistics and Computing* 17(3), 263–279.
- Johnstone, R. H., E. T. Chang, R. Bardenet, T. P. De Boer, D. J. Gavaghan, P. Pathmanathan, R. H. Clayton, and G. R. Mirams (2016). Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models? *Journal of Molecular and Cellular Cardiology* 96, 49–62.

- Lam, C. and J. Fan (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of Statistics* 37(6B), 4254.
- Lambert, B., M. Robinson, C. L. Lei, R. Creswell, M. Clerx, S. Ghosh, G. R. Mirams, S. Tavener, and D. J. Gavaghan (2020). Autoregressive errors for ordinary differential equation models. *preprint*.
- Lei, C. L., M. Clerx, K. A. Beattie, D. Melgari, J. C. Hancox, D. J. Gavaghan, L. Polonchuk, K. Wang, and G. R. Mirams (2019). Rapid characterization of hERG channel kinetics II: temperature dependence. *Biophysical Journal* 117(12), 2455–2470.
- Lei, C. L., M. Clerx, D. J. Gavaghan, L. Polonchuk, G. R. Mirams, and K. Wang (2019). Rapid characterization of hERG channel kinetics I: using an automated high-throughput system. *Biophysical Journal* 117(12), 2438–2454.
- Lei, C. L., M. Clerx, D. G. Whittaker, D. J. Gavaghan, T. P. De Boer, and G. R. Mirams (2020). Accounting for variability in ion current recordings using a mathematical model of artefacts in voltage-clamp experiments. *Philosophical Transactions of the Royal Society A* 378(2173), 20190348.
- Lewandowski, D., D. Kurowicka, and H. Joe (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis* 100(9), 1989–2001.
- Martínez, A. F. and R. H. Mena (2014). On a nonparametric change point detection model in markovian regimes. *Bayesian Analysis* 9(4), 823–858.
- Milstein, J. (1981). The inverse problem: estimation of kinetic parameters. In *Modelling of Chemical Reaction Systems*, pp. 92–101. Springer.
- Moles, C. G., P. Mendes, and J. R. Banga (2003). Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Research* 13(11), 2467–2474.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, Chapter 5. Chapman and Hall CRC.
- Paciorek, C. J. (2003). *Nonstationary Gaussian processes for regression and spatial modelling*. Ph. D. thesis, Carnegie Mellon University.
- Paciorek, C. J. and M. J. Schervish (2004). Nonstationary covariance functions for gaussian process regression. In *Advances in Neural Information Processing Systems*, pp. 273–280.
- Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer.
- Schäfer, J. and K. Strimmer (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology* 4(1).

- Silk, D., P. D. Kirk, C. P. Barnes, T. Toni, A. Rose, S. Moon, M. J. Dallman, and M. P. Stumpf (2011). Designing attractive models via automated identification of chaotic and oscillatory dynamical regimes. *Nature Communications* 2(1), 1–6.
- Stan Development Team (2016). Stan modeling language users guide and reference manual. *Technical report*.
- Wiener, N. (1950). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA.
- Williams, C. K. and C. E. Rasmussen (2006). *Gaussian processes for machine learning*, Volume 2. MIT press Cambridge, MA.
- Zhu, C., R. H. Byrd, P. Lu, and J. Nocedal (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* 23(4), 550–560.

**S1. Stationary AR(1) noise with Laplacian kernel**

Accurate inference for stationary non-IID noise can be achieved using the standard Laplacian kernel,

$$C(t_i, t_j) = \sigma^2 e^{-|t_i - t_j|/L}. \quad (27)$$

Here, we show the results of the method applied to a synthetic logistic growth time series with autoregressive order 1 (AR(1)) error terms. These results are shown in Figure S1. Panel (a) shows a synthetic noisy time series. The underlying model trajectory, labelled “Noise-free trajectory”, is calculated from a logistic growth model,

$$\frac{dy}{dt} = ry(1 - y/K). \quad (28)$$

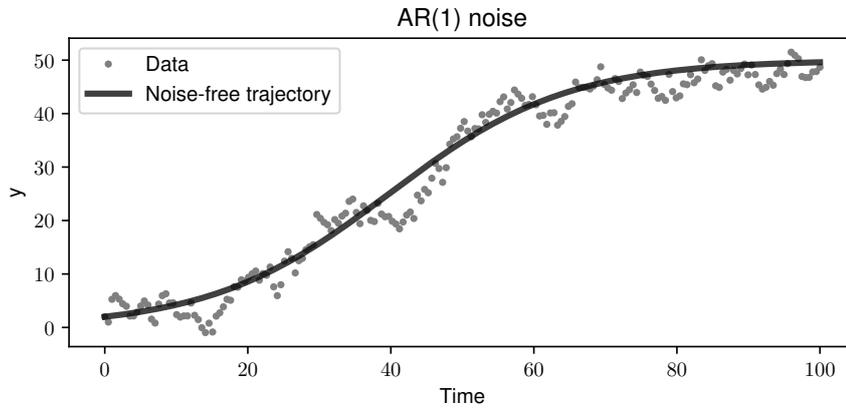
The AR(1) time series shows persistence in the error terms: the error term at any given time points depends both on a random fluctuation as well as the previous observation. Specifically, we model each error term  $\epsilon_i = y_i - f(t_i; \theta)$  according to:

$$\epsilon_i = \rho\epsilon_{i-1} + v_i, \quad (29)$$

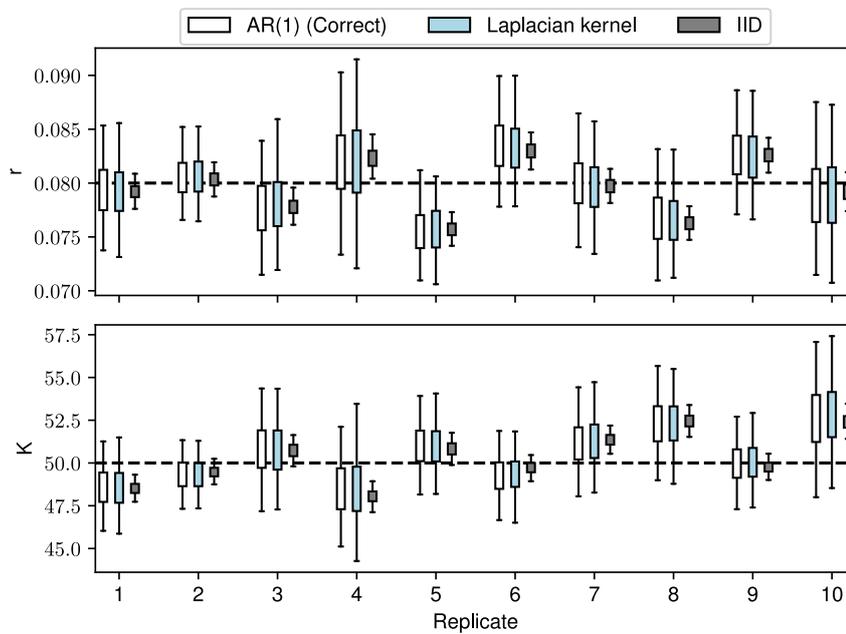
where  $v_i$  is Gaussian white noise,  $v_i \sim N(0, \sigma\sqrt{1 - \rho^2})$ . In these simulations, we used  $\rho = 0.8$  and  $\sigma = 3$ . Ten replicates of the time series with AR(1) noise were generated. For each time series, Bayesian inference for the parameters  $r$  and  $K$  was performed for each of three noise processes we consider: IID Gaussian with unknown variance (incorrectly specified), AR(1) with two unknown parameters (correctly specified), and the multivariate Gaussian likelihood with Laplacian kernel covariance. MCMC sampling was performed using three chains of the Haario Bardenet adaptive covariance algorithm, with a total of 20000 iterations in each chain (Haario et al., 2001; Johnstone et al., 2016). The first half of each chain was discarded as warm-up, and convergence was assessed using the Gelman  $\hat{R}$  statistic (Gelman et al., 2013). In Panel (b), the results of posterior inference for  $r$  and  $K$  are shown under the three noise processes, across 10 replicates. In each replicate, the bars indicate the central 95% of the posterior, while the green lines indicate the true values of the posterior. In each replicate, the first posterior with the correctly specified AR(1) noise process shows relatively high posterior uncertainty. The general multivariate normal noise process with Laplacian kernel reproduces the high level of posterior uncertainty in model parameters. By contrast, incorrectly specified IID assumption underestimates posterior uncertainty.

**S2. Non-stationary Laplacian kernel on blocked synthetic data**

This section shows an example of the GP non-stationary kernel method being applied to a synthetic time series with very sharp changes in the true noise parameters. The noise process had 5 regimes, and was used with a logistic growth ODE model. The first, third and fifth regimes had IID Gaussian noise with  $\sigma = 3$ , the second regime had AR(1) noise with  $\rho = 0.85$  and  $\sigma = 3$ , and the fourth regime had IID Gaussian noise with  $\sigma = 30$ . We found the MAP estimates of the non-stationary Laplacian kernel parameters, using Algorithm 2 for initialisation. In Figure S2, the results are shown. The top panel shows one replicate of the data and the MAP estimate of the model trajectory. In the bottom



(a)



(b)

Fig. S1: **Capturing AR(1) noise using a stationary Laplacian kernel.** Panel (a) shows a logistic growth time series with AR(1) noise. Ten replicates of the AR(1) time series were generated. Panel (b) shows the posterior distributions for logistic growth model parameters under three different assumptions for the noise process for each replicate. The boxes cover the central 50% posterior estimates, while the whiskers cover the central 95% posterior estimates. The dashed lines indicate true parameter values.

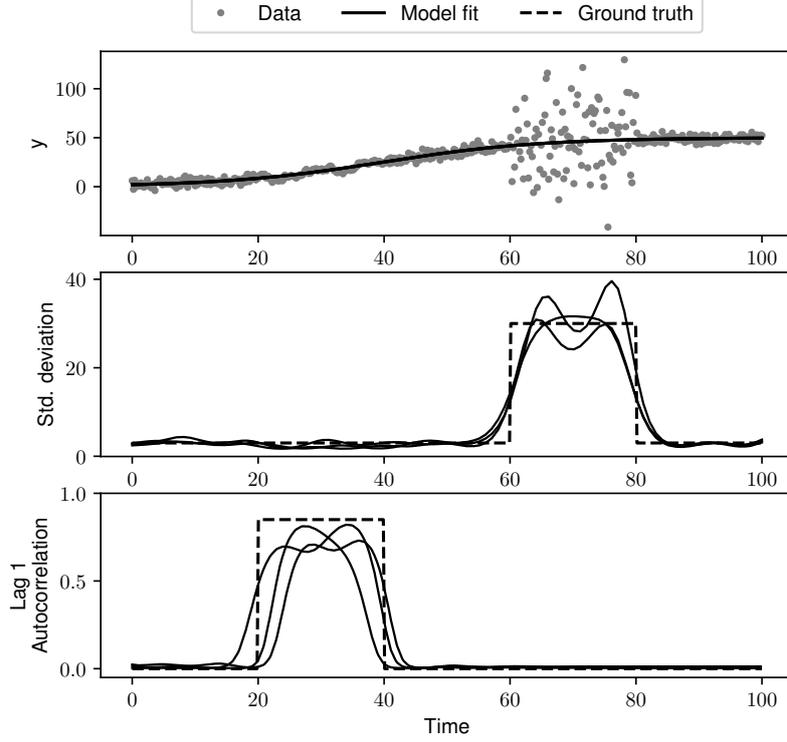


Fig. S2: **Non-stationary kernel method fit to blocked noise data.** This figure shows how the Gaussian processes in the non-stationary Laplacian kernel handle a noise series with blocks of different types of noise. The top plot shows a logistic growth time series with 5 blocks of different noise forms. In the middle and bottom plots, the true values of standard deviation and lag 1 autocorrelation are shown as dotted lines, while the inferred MAP estimates for standard deviation and lag 1 autocorrelation are shown in solid lines.

two panels, the inferred standard deviation and lag 1 autocorrelation are shown for three replicates. The GPs are unable to learn the sharp corners in the ground truth for standard deviation and autocorrelation, but they do reach a smooth approximation of the ground truth.

### S3. Blocked covariance prior distribution

In this section, we present some visualisations of the PPM prior used in the block covariance method. The prior over partitions, given by

$$p(\rho_N) = \left( \frac{N!}{k!} \frac{\prod_{i=1}^{k-1} (\phi + is)}{(\phi + 1)_{N-1\uparrow}} \prod_{j=1}^k \frac{(1-s)_{n_j-1\uparrow}}{n_j!} \right), \quad (30)$$

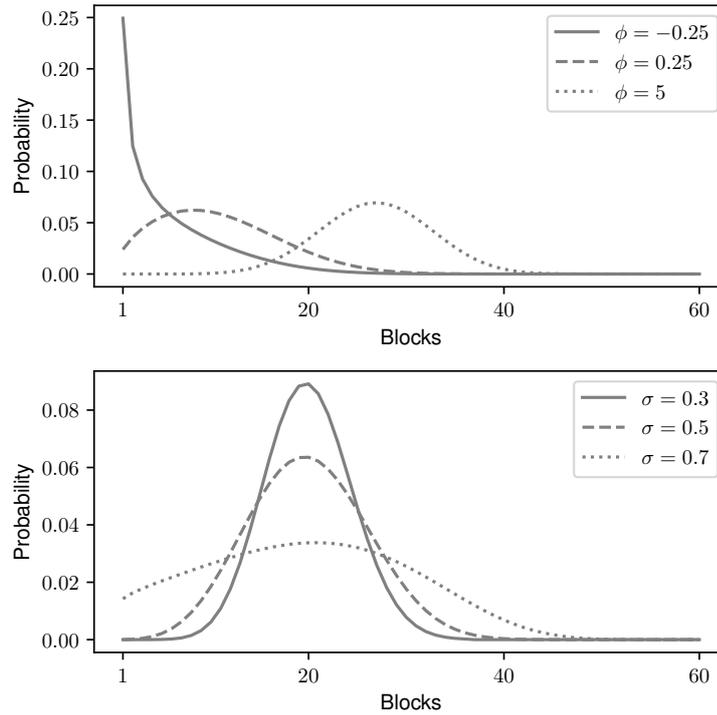


Fig. S3: **Marginal prior over number of blocks.** This figure shows slices of the marginal prior over the number of blocks used (Martínez and Mena, 2014). In the top panel,  $\sigma$  is fixed to 0.5. In the bottom panel, the mean number of blocks is fixed to 20.

does not admit direct interpretation or visualisation, but the marginal prior distribution over the number of blocks offers insight into (30) and its hyperparameters (Martínez and Mena, 2014). Some slices of the marginal distribution over the number of blocks used are shown in Figure S3. In all cases, the total number of time points  $N$  is fixed to 60. In the top panel,  $s$  is fixed to 0.5 and the prior is illustrated for three values of  $\phi$ . This shows how  $\phi$  works as a location parameter, shifting the prior mass to higher numbers of blocks as it increases. In the bottom panel, the mean number of blocks is fixed to 20 and the prior is shown for three values of  $s$ , which clearly serves as a shape parameter controlling the spread of the prior distribution.

**S4. MCMC algorithm for block covariance**

This section describes our split-merge-shuffle MCMC algorithm for inferring blocked covariance matrices. The algorithm closely follows the split-merge-shuffle algorithm for the conjugate case (Martínez and Mena, 2014), modified for non-conjugacy using ideas from the SAMS sampler (Dahl, 2005). We assume as input a parametric model  $f(t; \theta)$ , a positive definite kernel  $C_{\psi_j}$ , observed data  $\{y_i\}_{i=1}^N$  at time points  $\{t_i\}_{i=1}^N$ , partition prior hyperparameters  $s$  and  $\phi$ , with  $\psi_j$  indicating the full set of kernel parameters defining  $C$  in the  $j$ th block. Each data point is assigned to a block using indicator variables  $\{z_i\}_{i=1}^N$ . The steps for the sampler are provided in Algorithm 3.

To calculate  $\alpha_{\text{split}}$ , the acceptance probability for a proposed split, we must first propose a new value for  $\psi$  in the new block. Following Dahl (2005), we propose  $\psi_{\text{split}}$  according to a random walk, using a Gaussian proposal centred on the current value with a fixed variance. Letting  $\rho$  and  $\rho^*$  indicate the original and proposed split partitions, and  $\psi$  and  $\psi^*$  the original and proposed vectors of block kernel parameters, the acceptance ratio can then be calculated according to:

$$\alpha_{\text{split}} = \min \left( 1, \frac{p(\rho^*, \psi^* | y) q(\rho, \psi | \rho^*, \psi^*)}{p(\rho, \psi | y) q(\rho^*, \psi^* | \rho, \psi)} \right), \quad (31)$$

where  $p(\rho, \psi | y)$  denotes the posterior density evaluated at the partition  $\rho$  and block kernel parameters  $\psi$ , while  $q(\rho, \psi | \rho^*, \psi^*)$  is the probability of proposing  $\rho$  and  $\psi$  from  $\rho^*$  and  $\psi^*$ . The basic forward and reverse proposal probabilities for the split are obtained in Martínez and Mena (2014). Here, the forward probability must be multiplied by the proposal density used to obtain  $\psi_{\text{split}}$ .  $\alpha_{\text{merge}}$  indicates the acceptance probability for a proposed merge. The same eq. (31) applies, while in this case the reverse probability must be multiplied by the proposal density for  $\psi_{\text{split}}$ . The shuffle step is simpler as the dimension remains unchanged; the proposal probabilities for shuffling are taken

from Martínez and Mena (2014).

---

**Algorithm 3:** MCMC sampler for block covariance
 

---

**Input:** A parametric model  $f(t; \theta)$ , a positive definite kernel function  $C_{\psi_j}$ , observed data  $\{y_i\}_{i=1}^N$  at time points  $\{t_i\}_{i=1}^N$ . Initial values for the following parameters:  $\theta$ , assignments  $\{z_i\}_{i=1}^N$  of each time point to blocks,  $\psi_j$  for each initial block  $j$ , and partition prior hyperparameters  $s$  and  $\phi$ . Desired number of MCMC iterations  $N_{\text{MCMC}}$ .

**Output:** MCMC samples for  $\theta$ ,  $z$ ,  $s$ ,  $\phi$ , and  $\psi$ .

$K \leftarrow$  initial number of blocks;

**for**  $j = 1 \dots K$  **do**

$n_j \leftarrow \sum_{i=1}^N \mathbf{1}(z_i = j)$ ; // Count number of points in each block

**end**

**for**  $m = 1 \dots N_{\text{MCMC}}$  **do**

Update  $\theta$  via one adaptive covariance MCMC step;

Update  $s$  and  $\phi$  via standard MH steps;

**for**  $j = 1 \dots K$  **do**

Update  $\psi_j$  via standard MH steps;

**end**

Draw  $u$  from uniform(0, 1);

**if**  $K = 1$  or  $u < 0.25$  **then**

$z, \psi, K \leftarrow \text{split}(z, \psi, K, n)$ ; // Run the split routine (Alg. 4)

**else**

$z, \psi, K \leftarrow \text{merge}(z, \psi, K, n)$ ; // Run the merge routine (Alg. 5)

**end**

**for**  $j = 1 \dots K$  **do**

$n_j \leftarrow \sum_{i=1}^N \mathbf{1}(z_i = j)$ ; // Recount number of points in each block

**end**

$z \leftarrow \text{shuffle}(z, K, n)$ ; // Run the shuffle routine (Alg. 6)

**end**

---



---

**Algorithm 4:** Split proposal step
 

---

**Input:** Assignments  $z$ , Kernel block parameters  $\psi$ , number of blocks  $K$ , number of points in each block  $n$ .

**Output:** Updated assignments  $z$ , block parameters  $\psi$  and number of blocks  $K$  after one Metropolis-Hastings split step.

**Function**  $\text{split}(z, \psi, K, n)$ :

Randomly select a block  $j$  from  $\{j : 1 \leq j \leq K, n_j > 1\}$ ;

Randomly select an index  $l$  from  $\{1, \dots, n_j - 1\}$ ;

Draw  $u'$  from uniform(0, 1);

**if**  $u' < \alpha_{\text{split}}$  **then**

**for**  $i = 1 \dots N$  **do**

**if**  $i > l + \sum_{j'=1}^{j-1} n_{j'}$  **then**

$z_i \leftarrow z_i + 1$ ;

$\psi \leftarrow (\psi_1, \dots, \psi_j, \psi_{\text{split}}, \psi_{j+1}, \dots, \psi_K)$ ;

$K \leftarrow K + 1$ ;

**return**  $z, \psi, K$ ;

---



---

**Algorithm 5:** Merge proposal step
 

---

**Input:** Assignments  $z$ , kernel block parameters  $\psi$ , number of blocks  $K$ , number of points in each block  $n$ .

**Output:** Updated assignments  $z$ , block parameters  $\psi$  and number of blocks  $K$  after one Metropolis-Hastings merge step.

**Function**  $\text{merge}(z, \psi, K, n)$ :

Randomly select a block  $j$  from  $\{j : 1 \leq j \leq K - 1\}$ ;

Draw  $u'$  from uniform(0, 1);

**if**  $u' < \alpha_{\text{merge}}$  **then**

**for**  $i = 1 \dots N$  **do**

**S5. hERG Hodgkin-Huxley model parameter priors**

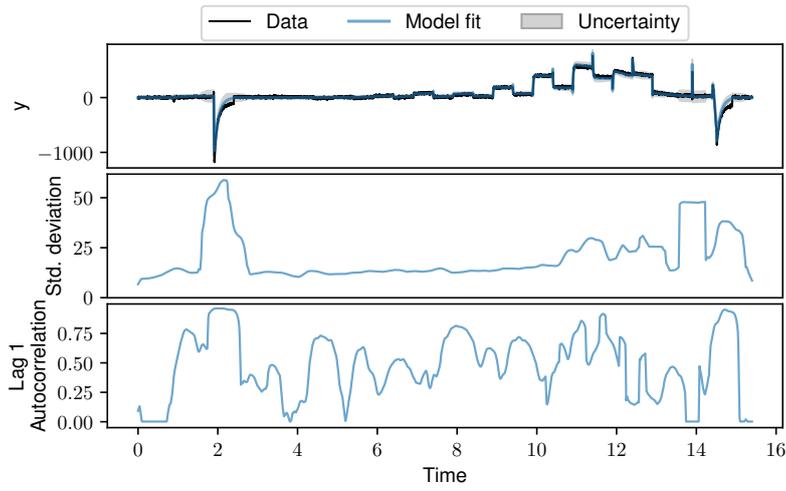
In this section, we list the priors used for the 9 log-transformed model parameters in the hERG model introduced in §6.1.

Parameter	Prior
$g_{Kr}$	$\mathcal{N}(10.5, 1.0)$
$p_1$	$\mathcal{N}(-2.5, 3.0)$
$p_2$	$\mathcal{N}(4.5, 1.0)$
$p_3$	$\mathcal{N}(-3.5, 1.5)$
$p_4$	$\mathcal{N}(4.0, 0.5)$
$p_5$	$\mathcal{N}(4.5, 0.5)$
$p_6$	$\mathcal{N}(3.0, 1.5)$
$p_7$	$\mathcal{N}(2.0, 0.5)$
$p_8$	$\mathcal{N}(3.5, 0.5)$

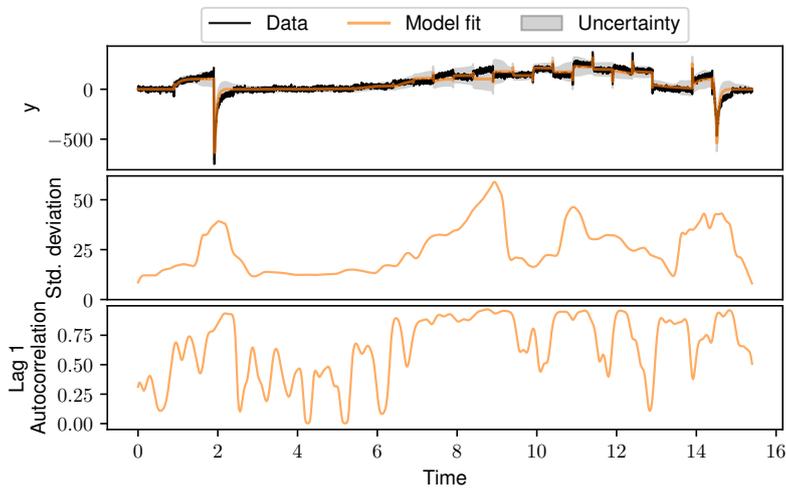
Table S1: **hERG model prior parameters.** This table contains the prior distributions used for each parameter in the hERG model. For each parameter, the prior is a normal distribution with the mean and standard deviation given in the table.

**S6. hERG data and model fits**

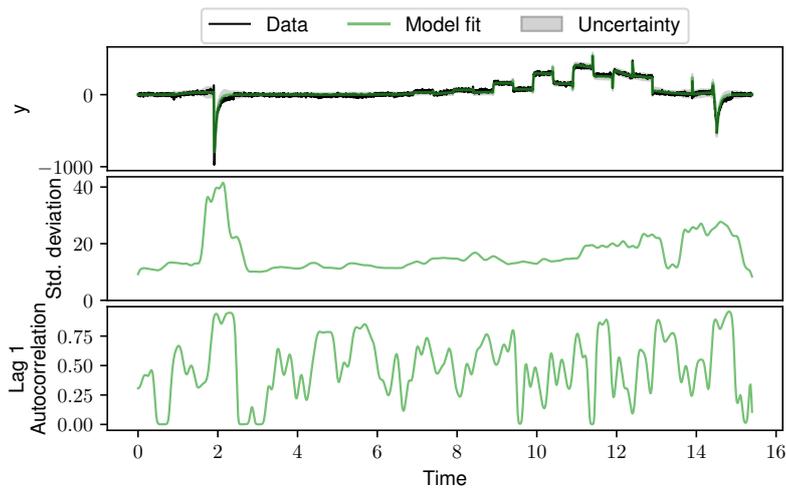
This section contains the time series data and model fit for six hERG cells. The main features of the Gaussian process fits are two large spikes in standard deviation appearing around times  $t = 2$  and  $t = 14.5$ . While there is some variation from cell to cell, these spikes correspond to regions of the time series with rapid drops in current. Autocorrelation is fairly high throughout the time series, with wide fluctuations. Cells A04 and A07 exhibit another peak in standard deviation and autocorrelation around  $t = 8$ ; this corresponds to a region of obvious discrepancy between the model and the data.



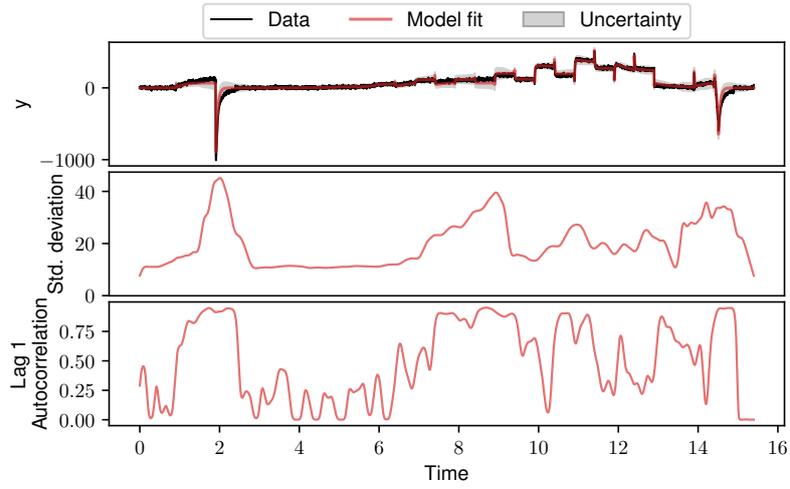
(a) A01



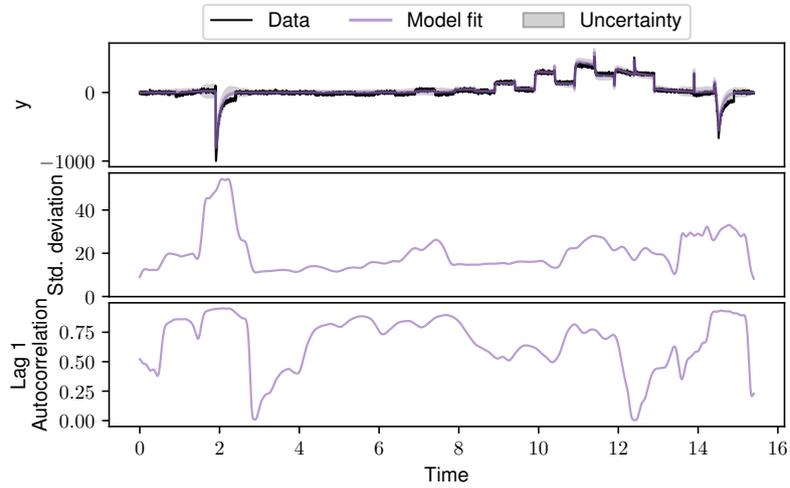
(b) A04



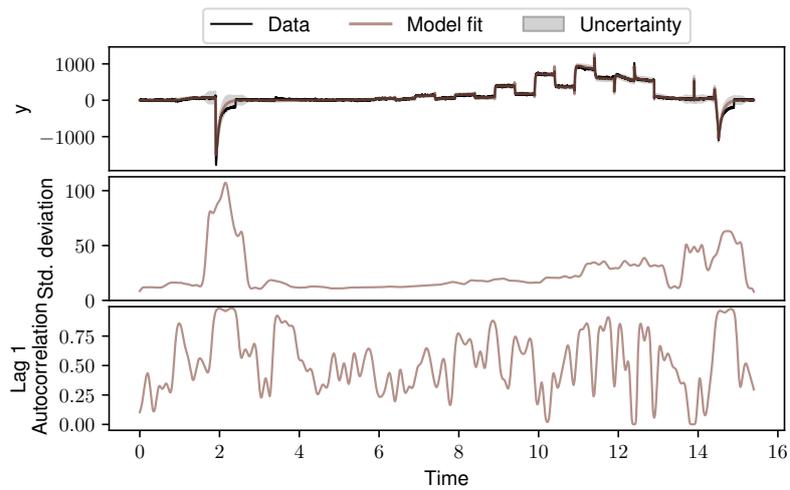
(c) A06



(d) A07



(e) A09



(f) A11

Fig. S4: **Non-stationary Laplacian kernel noise models fit to hERG data.** This figure shows the data and model fit (top panel), and the MAP estimate standard deviation and lag 1 autocorrelation over time (second and third panels) inferred by the non-stationary Laplacian kernel noise model for each of six cells.