Calibrating Black-Box Probabilistic Numerical Methods

Juntao Chen — *Newcastle University, UK*
Markus Michael Rau — *Newcastle University, UK*
Chris. J. Oates — *Newcastle University, UK*

Abstract

Black-box probabilistic numerical methods offer the tantalising prospect of endowing any (scalar) output from any (consistent) numerical code with a Bayesian credible interval. The idea is to construct a dataset containing simulations of increasing precision, and to cast extrapolation of these data to the infinite precision limit as a prediction task. However, calibrating probabilistic predictions is challenging when working with a small dataset. For codes with multiple outputs, treating each (scalar) output independently can result in large variation in credible intervals between outputs whose numerical accuracy ought to be similar. A principled solution is to formulate a multivariate prediction task, but this requires a statistical model capable of describing the (possibly complex) multivariate phenomenon being simulated, which goes against the spirit of the black-box framework. This paper proposes and analyses a simple approach to couple together related prediction tasks while preserving the simplicity of the black-box framework.

Proceedings of the 2nd International Conference on Probabilistic Numerics (ProbNum) 2026, Lappeenranta, Finland. PMLR Volume 341.

1 Introduction

Numerical simulations are essential across science and engineering, enabling prediction, optimisation, and design for complex systems that are difficult or impossible to study experimentally. Their value, however, depends on the trustworthiness of their output: numerical errors arising from discretisation, round-off, or algorithmic instability can produce precise-looking but misleading results, undermining the decisions based upon them.

Probabilistic numerics (PN) is concerned with probabilistic uncertainty quantification for numerical tasks, as has seen success at foundational numerical tasks such as integration and solving linear systems of equations (Hennig et al., 2022). However, this has been achieved by re-inventing numerical methods from the ground-up, an approach which cannot easily scale to complex numerical codes that incorporate multiple different aspects of numerical approximation. Further, one cannot simply "replace all instances of classical numerical methods with PN methods" within a pipeline of code and expect the probabilistic output to be meaningful, as this requires specifying an appropriate joint distribution over all unknowns, which is difficult (Cockayne et al., 2019). As such, there remains a challenge to take the PN approach to large numerical codes.

The *black-box probabilistic numerics* (BBPN) approach introduced by Teymur et al. (2021) addresses a central challenge in PN by treating a (consistent) numerical method as a "black box" and modelling each (scalar) output $f(\mathbf{x})$ as a function of the numerical tolerances or step sizes $\mathbf{x} \in [0, \infty)^d$, with the aim of inferring the limiting quantity $f(\mathbf{0})$ corresponding to the exact quantity of interest. In this framework the function f is modelled as a realisation of a stochastic process, so that conditioning on a finite collection of evaluations $\mathcal{D} = \{f(\mathbf{x}_i)\}_{i=1}^n$ provides interpretable probabilistic predictions, beyond the observed resolution regime, for $f(\mathbf{0})|\mathcal{D}$. This perspective was placed on a rigorous theoretical footing by Oates et al. (2024), who formalised the approach as *Probabilistic Richardson Extrapolation* (PRE). In the particular case of a Gaussian process (GP), which the authors referred to as *Gauss–Richardson extrapolation* (GRE), the authors showed that the conditional mean $\mathbb{E}[f(\mathbf{0})|\mathcal{D}]$ can be asymptotically more accurate than the original numerical method, in close analogy with classical Richardson extrapolation (Richardson, 1911; Richardson and Gaunt, 1927). However, for the purposes of this paper the important point is that BBPN enables principled uncertainty quantification while remaining compatible with complex legacy numerical codes, requiring only black-box access to the numerical output.

This paper is concerned with uncertainty quantification for *multivariate* solver output within the BBPN framework. This setting is important since simulation output is often rich and structured. For illustration, consider the dynamical system of Lorenz (1963). Numerically integrating this system forward with time step $x \in (0, \infty)$ produces output $\{\mathbf{u}(t_i, x)\}_{i=1}^T \subset \mathbb{R}^3$ representing an approximation to the true state of the system at discrete time points $\{t_i\}_{i=1}^T$. Throughout, x denotes a discretisation parameter to be taken toward 0 (here, the time step of the integrator), while t indexes the components of the multivariate output (here, the simulation times); this convention is used consistently in what follows. Though in principle we could take each scalar component of this output and apply BBPN using GPs to perform uncertainty quantification for the true trajectory of the Lorenz system, i.e. $f(x) = u_j(t_i; x)$, this necessitates learning of $3T$ stochastic processes, each from a limited dataset. It is well-known that hyper-parameter estimates are unreliable at small dataset sizes, and this

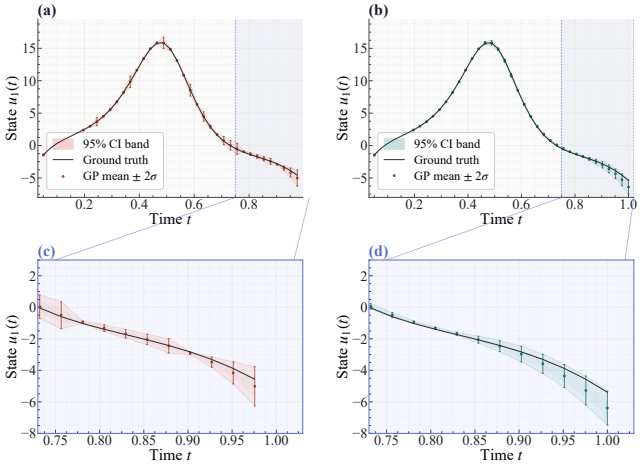


Figure 1: **Independent versus sequential multi-task GP extrapolation on the Lorenz system.** Panels (a) and (b) show the trajectory of the first coordinate $u_1(t)$ with 95% credible intervals; panels (c) and (d) zoom into the extrapolation region ($t \geq 0.75$). The independent GP underestimates predictive uncertainty beyond the training window, while the sequentially regularised model maintains well-calibrated credible intervals.

manifests as erratic variation in predictive error bars under this `independent` approach (left panel of Figure 1). It is important to emphasise that the Lorenz system serves only to illustrate multivariate simulator output; for numerically integrating differential equations, dedicated PN methods should be preferred (Hennig et al., 2022). Our motivation is to develop BBPN for complex multivariate simulators that do not easily admit a PN treatment.

In principle multivariate simulator output can be handled natively within the BBPN framework by constructing an augmented stochastic process model for $f(\mathbf{x}, t)$, where $t \in \mathcal{T}$ indexes the different (scalar) components of the simulator output. This approach was explored briefly in Oates et al. (2024, Section 2.9). Given a dataset $\mathcal{D} = \{f(\mathbf{x}_i, \cdot)\}_{i=1}^n$, the (multivariate) conditional $f(\mathbf{0}, \cdot)|\mathcal{D}$ represents a multivariate predictive distribution for all (scalar) quantities of interest. However, we find this approach unsatisfactory, as building a suitable stochastic process model for $f(\mathbf{x}, \cdot) : \mathcal{T} \to \mathbb{R}$ requires *a priori* knowledge of the simulator output that goes against the spirit of a "black-box" framework. For the Lorenz example, this amounts to building a stochastic process to model the solution of the Lorenz system, which is not entirely straightforward; this could be essentially impossible for numerical simulators producing complex structured output. As such, it would be desirable to find an alternative approach to extend BBPN to multivariate simulator output, which *deliberately avoids* constructing a joint probability distribution over the variables indexed by $\mathcal{T}$.

1.1 Our Contribution

This paper proposes a simple-but-useful heuristic to extend BBPN to multivariate output. Our heuristic is applicable in settings where the index set $\mathcal{T}$ can be ordered; then we may identify $\mathcal{T} = \{1, \ldots, T\}$.

Further, we assume that the outputs $f(\cdot, t)$ and $f(\cdot, s)$ are statistically similar when $t \approx s$. The reasonableness of this assumption depends on the context: for complex physical simulations in which t indexes space or time (or both), the assumption is typically underwritten by physical constraints, such as continuity of the continuum quantity of interest—this is the case for the Lorenz example, where t indexes the (ordered) times at which the system is simulated. For problems of a more discrete nature, such as the eigenvalue computation of Section 4.3, it is *a priori* less clear whether the assumption can hold, even approximately, though our empirical results in that setting are encouraging. There is no reason to expect our sequential approach to perform well when this assumption fails to (at least approximately) hold; this limitation is discussed further in Section 5.

Our main idea is simple: the miscalibration of `independent` predictions stems from unreliable estimation of GP hyper-parameters from the small per-task datasets, so, assuming a GP model is being used within BBPN, we propose to regularise the hyper-parameters across the task index—encouraging the hyper-parameters at task $t+1$ to take values similar to those at task t—thereby borrowing statistical strength between related tasks without ever constructing a joint model over $\mathcal{T}$.

This approach is straight-forward to implement and has complexity $O(n^3 T)$, linear in the cardinality T of $\mathcal{T}$. An example of the output from the `sequential` method is shown in the right panel of Figure 1; compared to `independent` predictions, our predictive error bars vary more smoothly and are better-calibrated. Though simple, we contend that this work represents a necessary step toward making BBPN a practical solution for quantifying numerical uncertainty in complex multivariate simulator output.

1.2 Related Work

Statistical models for multivariate functional data are well-established. Focusing on GPs, notable contributions include *multi-task* GPs (e.g. Bonilla et al., 2007) which, in our notation, construct GP models for $f(\mathbf{x}, t)$ over $[0, \infty)^d \times \mathcal{T}$. This approach is equivalent to minimal norm interpolation in vector-valued reproducing kernel Hilbert spaces (Carmeli et al., 2010; Alvarez et al., 2012), and also goes under the names of *multi-output* GP (MOGP) and *co-kriging* (Wackernagel, 2003). Extensions of multi-task GPs to increase expressivity (e.g. Wilson et al., 2012) and to accommodate multi-resolution data (e.g. Hamelijnck et al., 2019) have also been considered. Seeking a compromise between joint modelling and independent estimation, the *semiparametric latent factor model* (SLFM) of Teh et al. (2005) maintains a small number of latent GPs $\{g_i(\mathbf{x})\}_{i=1}^p$ and treats the observable as linear combinations

$$f(\mathbf{x}, t) = c_1(t) g_1(\mathbf{x}) + \cdots + c_p(t) g_p(\mathbf{x}) \qquad (1)$$

where both the coefficient functions $c_i(t)$ and the GPs $g_i(\mathbf{x})$ are to be learned. Our main objection to the above approaches in the setting of BBPN is that specifying a suitable statistical dependence model over $\mathcal{T}$ can be difficult; a generic (continuous) covariance kernel is likely to over-smooth parts of complex structured

numerical output. A secondary objection is that the computational overhead associated with joint modelling can increase substantially when $\mathcal{T}$ is a large set, except perhaps if restrictive assumptions are made on the form of the joint covariance kernel, or if more sophisticated computational approximations are used (e.g. Da et al., 2019).

It is worth emphasising what distinguishes BBPN from generic multivariate GP regression: the mean and covariance structure of BBPN are constructed so that the convergence of the posterior mean to the quantity of interest is *provably faster* than the convergence of the data on which it is trained, provided sufficient smoothness can be exploited (Oates et al., 2024; Blinded, 2026). Such convergence acceleration does not occur automatically for a generic multivariate GP model. These theoretical guarantees concern the posterior mean and were established in earlier work; the focus of the present paper is instead on the posterior *variance*, and how it can be calibrated in practice for multivariate output.

More broadly, there is a rich literature on *multi-fidelity modelling* (see e.g. Peherstorfer et al., 2018), which approximates high-fidelity simulator output from low-fidelity output; these methods aim for general usage and do not explicitly extrapolate to the continuum limit, so for PN, where the continuum limit is the principal goal, alternative frameworks such as BBPN are required.

An equivalent perspective is that we wish to solve multiple numerical tasks, indexed by $\mathcal{T}$, using PN. Bayesian quadrature for multiple related integrals was studied in Xi et al. (2018), where a joint GP model was constructed over the collection of integrands. Solving multiple related linear systems was addressed through the lens of PN in Hegde and Cockayne (2025), who assumed an ordered sequence of tasks and used information from previous solves to build a preconditioner for the next task. Our motivation instead derives from complex numerical simulations for which such "white-box" PN methods cannot easily be developed.

2 Background

This section presents necessary background, starting with the *sparse* PRE approach of Blinded (2026) for estimating a single (scalar) quantity of interest, in Section 2.1, and recalling the role of restricted marginal likelihood for calibrating uncertainty in Section 2.2. The challenge of extending sparse probabilistic Richardson extrapolation (SPRE) to multivariate output is discussed in Section 2.3.

2.1 Sparse PRE

For BBPN, in this work we employ the recent *sparse* PRE (SPRE) method of Blinded (2026). Let $\mathbf{0} \in A = \{\boldsymbol{\alpha}_i\}_{i=1}^m \subset \mathbb{N}_0^d$ where we adopt the convention that $\boldsymbol{\alpha}_1 = \mathbf{0}$. For a numerical method f producing a single (scalar) output, the idea of SPRE is to model f as a GP, whose mean function

$$\mathbb{E}[f(\mathbf{x})|\{\beta_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in A\}] = \sum_{\boldsymbol{\alpha}\in A} \beta_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \tag{2}$$

depends on unknown coefficients $\beta_{\boldsymbol{\alpha}}$, which are in turn assigned an improper flat prior and integrated out, while the covariance function

$$\mathbb{C}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}')$$

is determined by a symmetric and positive definite kernel k to be specified. Compared to a generic GP, the key distinguishing feature of SPRE is that the index set A should be selected such that $f(\mathbf{x})$ admits a Taylor expansion around $\mathbf{0}$ involving only the monomials indexed by A. If this can be achieved, then the conditional mean $\mathbb{E}[f(\mathbf{0})|\mathcal{D}]$ of SPRE is provably more accurate than the data on which it was trained. For complex codes which do not admit a straightforward error analysis, the set A can be treated as an additional hyper-parameter of the GP and learned. Compared to the earlier GRE construction of Oates et al. (2024), the SPRE construction is both simpler and requires fewer data for provable convergence acceleration, especially when the index set A is sparse; we defer to Blinded (2026) for further detail.

Following standard calculations[1] the posterior mean and variance can be explicitly computed. Indeed, let

$$\mathbf{V}_A = \begin{bmatrix} \mathbf{x}_1^{\boldsymbol{\alpha}_1} & \dots & \mathbf{x}_1^{\boldsymbol{\alpha}_m} \\ \vdots & & \vdots \\ \mathbf{x}_n^{\boldsymbol{\alpha}_1} & \dots & \mathbf{x}_n^{\boldsymbol{\alpha}_m} \end{bmatrix}$$

and also let $\mathbf{K} := [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^n$, $\mathbf{f} := [f(\mathbf{x}_i)]_{i=1}^n$, $\mathbf{k}(\mathbf{x}) := [k(\mathbf{x}_i, \mathbf{x})]_{i=1}^n$, and $\mathbf{v}_A(\mathbf{x}) := [\mathbf{x}^{\boldsymbol{\alpha}_i}]_{i=1}^m$. Then, we have that

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}^*)|\{f(\mathbf{x}_i)\}_{i=1}^n] &= \mathbf{k}(\mathbf{x}^*)^\top \mathbf{K}^{-1}\mathbf{f} + \mathbf{r}_A(\mathbf{x}^*)^\top \hat{\boldsymbol{\beta}}_A \\ \mathbb{V}[f(\mathbf{x}^*)|\{f(\mathbf{x}_i)\}_{i=1}^n] &= k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*) \\ &\quad + \mathbf{r}_A(\mathbf{x}^*)^\top (\mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{V}_A)^{-1}\mathbf{r}_A(\mathbf{x}^*)\end{aligned}$$

where

$$\begin{aligned}\mathbf{r}_A(\mathbf{x}) &:= \mathbf{v}_A(\mathbf{x}) - \mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) \\ \hat{\boldsymbol{\beta}}_A &:= (\mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{V}_A)^{-1}\mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{f}.\end{aligned}$$

In particular we are interested in prediction at $\mathbf{x}^* = \mathbf{0}$, where under our convention $\mathbf{v}_A(\mathbf{0}) = \mathbf{e}_1$. Note that $\{\mathbf{x}_i\}_{i=1}^n$ is required to be $\mathcal{P}_A$-unisolvent in order for $\mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{V}_A$ to be inverted.

2.2 Calibrating Uncertainty

Since the polynomial coefficients $\beta_{\boldsymbol{\alpha}}$ are assigned an improper flat prior, we do not have a well-defined marginal likelihood. Instead we use the restricted marginal likelihood (REML) to select a suitable kernel k for SPRE (Harville, 1974):

$$p_{\text{REML}}(\mathcal{D} \mid \boldsymbol{\theta}) \propto \exp\Big(-\tfrac{1}{2}\log|\mathbf{K}| - \tfrac{1}{2}\log|\mathbf{V}_A^\top \mathbf{K}^{-1}\mathbf{V}_A| - \tfrac{1}{2}\mathbf{f}^\top \mathbf{P}_A \mathbf{f}\Big)$$

[1] See for instance Section 2.7 of Rasmussen and Williams (2006).

where $\mathbf{P}_A = \mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{V}_A(\mathbf{V}_A^\top\mathbf{K}^{-1}\mathbf{V}_A)^{-1}\mathbf{V}_A^\top\mathbf{K}^{-1}$ is the projection matrix that orthogonalises the data with respect to the polynomial trend.

For the purposes of this work we consider k to be of the form

$$k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \sigma^2\phi\left(\frac{\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right)$$

for hyper-parameters $\boldsymbol{\theta} = (\log(\sigma^2), \log(\ell)) \in \mathbb{R}^2$ and a radial basis function $\phi : [0, \infty) \to [0, \infty)$, though more sophisticated kernels could also be considered (Rasmussen and Williams, 2006).

2.3 Extension to Multivariate Output

Extending the SPRE approach to multivariate output $f(\mathbf{x}, t)$ is not entirely straight forward. A simple approach is to treat each index $t \in \mathcal{T}$ in turn and fit an `independent` GP, calibrating the kernel parameters $\boldsymbol{\theta}_t$ in each case using REML as described in Section 2.2 based on the t-specific dataset

$$\mathcal{D}_t := \{f(\mathbf{x}_i, t)\}_{i=1}^n.$$

However, the variability of the hyper-parameter estimates obtained using REML leads to high variability in predictive credible intervals. To illustrate this, we consider predicting the true solution of the Lorenz system at different time points t, in each case extrapolating from data obtained using a numerical integrator run at a small number of different temporal resolutions (full details are contained in Section 4.2). The results are illustrated in the left panel of Figure 1; predictions made at consecutive time points can differ substantially in terms of the length of their predictive credible interval. Further, we will show in Section 4 that these `independent` credible intervals are over-confident in general.

On the other hand, constructing a joint model for $f(\mathbf{x}, t)$ could be extremely difficult; the complex simulators that motivate this work return structured output which is not easily captured by a "generic" GP, and moreover the computational burden of inference can increase dramatically when working with a joint model. This motivates the development of a simple-but-useful heuristic strategy for multi-output BBPN, which we present next.

3 Methods

This section presents a simple strategy based on modelling the datasets $\mathcal{D}_t$ ($t \in \mathcal{T} = \{1, \ldots, T\}$) as conditionally independent given the hyper-parameters $\boldsymbol{\theta}_{1:T} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T)$ of the kernel; this approach obviates the need to specify statistical dependencies over $\mathcal{T}$, and the joint log-REML of the full dataset is

$$\mathcal{L}(\boldsymbol{\theta}_{1:T}) = \sum_{t=1}^{T} \log p_{\text{REML}}(\mathcal{D}_t \mid \boldsymbol{\theta}_t). \tag{3}$$

Then, we aim to "borrow strength" across the outputs indexed in $\mathcal{T}$ by forcing hyper-parameters $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_s$ to take similar values when considering closely related outputs $t, s \in \mathcal{T}$.

3.1 Global Hyper-Parameters

The simplest way to borrow strength is to learn a single `global` vector of hyper-parameters $\boldsymbol{\theta}$ so that $\boldsymbol{\theta}_{1:T} = (\boldsymbol{\theta}, \ldots, \boldsymbol{\theta})$. Though this pooled optimisation has $O(T)$ complexity per gradient evaluation—the same asymptotic order as the `sequential` method developed below—asymptotic order alone does not determine practical cost, since the optimisation structure, constant factors, and scope for parallelisation differ between the two formulations. Here we instead consider an embarrassingly parallel alternative, which first fits hyper-parameters $\boldsymbol{\theta}_{1:T}$ by maximising the REML (3), producing an estimator $\hat{\boldsymbol{\theta}}_{1:T}$, and then combines these hyper-parameters to obtain a single parameter vector $\hat{\boldsymbol{\theta}}$. As a baseline, we consider the simplest combination strategy via an arithmetic mean

$$\hat{\boldsymbol{\theta}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\boldsymbol{\theta}}_t.$$

Finally, predictions are made at each $t \in \mathcal{T}$ using SPRE with hyper-parameters $\hat{\boldsymbol{\theta}}$. This `global` heuristic will be employed as a baseline for the `sequential` method we develop next.

3.2 Sequential Learning

The `global` approach has the potential to reduce estimator variance in the small data setting that is typical for BBPN, but a single fixed choice of $\boldsymbol{\theta}$ could lead to some predictions being over-confident while others are under-confident. To address this possibility, we must allow the $\boldsymbol{\theta}_t$ to be distinct. Our proposal is to enforce similarity between hyper-parameters $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_s$ when $t \approx s$, recalling that the index $t \in \mathcal{T}$ is selected such that $f(\cdot, t)$ and $f(\cdot, s)$ are expected to be statistically similar when $t \approx s$. In practice we achieve this using a hyper-prior for $\boldsymbol{\theta}_{1:T}$ which views $(\boldsymbol{\theta}_t)_{t=1}^T$ as a Markov chain, and then perform maximum a posteriori (MAP) estimation for $\boldsymbol{\theta}_{1:T}$. Intuitively, the prior acts as a shrinkage device: each $\boldsymbol{\theta}_t$ is informed not only by its own dataset $\mathcal{D}_t$ but also by the datasets of neighbouring tasks, with the strength of this coupling controlled by the prior. The approach may thus be understood as a continuous interpolation between the `independent` and `global` extremes, with the appropriate degree of information-sharing learned from the data.

3.2.1 Gaussian Random Walk Prior

To regularise the kernel hyper-parameters across $t \in \mathcal{T}$, we impose a first-order Gaussian random walk prior on the state trajectory $\boldsymbol{\theta}_{1:T}$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{4}$$

where $\mathbf{Q} = \text{diag}(\tau_\sigma^2, \tau_\ell^2)$. Then, with an improper flat prior on the initial state, $p(\boldsymbol{\theta}_1) \propto 1$, the induced prior

over $\boldsymbol{\theta}_{1:T}$ is a chain-structured Gaussian Markov random field:

$$p(\boldsymbol{\theta}_{1:T}) \propto \exp\left(-\frac{1}{2}\sum_{t=1}^{T-1}(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)^\top \mathbf{Q}^{-1}(\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t)\right).$$

The diagonal structure of $\mathbf{Q}$ allows the evolution of the log-amplitude $\log\sigma_t$ and log-lengthscale $\log\ell_t$ to be independent.

3.2.2 Maximum A Posteriori Estimation

Let $\lambda_\sigma = (2\tau_\sigma^2)^{-1}$ and $\lambda_\ell = (2\tau_\ell^2)^{-1}$. Combining the joint log-REML (3) with the Gaussian random walk prior (4) yields the MAP estimator of the hyperparameter trajectory:

$$\hat{\boldsymbol{\theta}}_{1:T} = \arg\max_{\boldsymbol{\theta}_{1:T}} \Big[\mathcal{L}(\boldsymbol{\theta}_{1:T}) - \lambda_\sigma \sum_{t=1}^{T-1}(\theta_{1,t+1}-\theta_{1,t})^2 - \lambda_\ell \sum_{t=1}^{T-1}(\theta_{2,t+1}-\theta_{2,t})^2\Big] \tag{5}$$

where it can be seen that λ_σ and λ_ℓ control the penalisation of temporal increments. In the limiting regimes, $\lambda_\sigma, \lambda_\ell \to 0$ recover `independent` per-time-point estimation, while $\lambda_\sigma, \lambda_\ell \to \infty$ recovers the `global` hyperparameter estimation approach.

For the purpose of this work, optimisation is carried out over $\boldsymbol{\theta}_{1:T} \in \mathbb{R}^{2T}$ using L-BFGS-B (Byrd et al., 1995), with analytic gradients for the REML. Optimisation was initialised from the `independent` estimator, which maximises only the REML.

3.2.3 Selection of λ_σ and λ_ℓ

The MAP objective (5) involves regularisation parameters $\boldsymbol{\lambda} = (\lambda_\sigma, \lambda_\ell)$, controlling the smoothness of the log-amplitude trajectory and the log-lengthscale trajectory. Since these components may exhibit different degrees of variability, we treat λ_σ and λ_ℓ as distinct tuning parameters and select them jointly by blocked B-fold cross-validation over the task index, to assess out-of-task generalisation. For this, we partition $\mathcal{T}$ into B contiguous, non-overlapping folds $\mathcal{T}_1, \ldots, \mathcal{T}_B$ of approximately equal size. For $\boldsymbol{\lambda}$ in a finite candidate set $\Lambda_\sigma \times \Lambda_\ell$, and for each fold b, the penalised MAP problem (5) is solved on the training indices $\mathcal{T} \setminus \mathcal{T}_b$, yielding estimates $\{\hat{\boldsymbol{\theta}}_t\}_{t\notin\mathcal{T}_b}$. For indices $t \in \mathcal{T}_b$, the corresponding hyper-parameters $\hat{\boldsymbol{\theta}}_t$ are obtained by linear interpolation in the index variable between the nearest training indices $s < t < u$,

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_s + \frac{t-s}{u-s}\left(\hat{\boldsymbol{\theta}}_u - \hat{\boldsymbol{\theta}}_s\right). \tag{6}$$

Under the first-order Gaussian random walk prior, the conditional expectation $\mathbb{E}[\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_s, \boldsymbol{\theta}_u]$ is linear in t, so (6) coincides with the posterior mean implied by that prior. The held-out score for fold b is

$$S_b(\boldsymbol{\lambda}) = \frac{1}{|\mathcal{T}_b|}\sum_{t\in\mathcal{T}_b} \log p_{\mathrm{REML}}\left(\mathcal{D}_t \mid \hat{\boldsymbol{\theta}}_t\right).$$

Averaging over all B folds yields

$$S(\boldsymbol{\lambda}) = \frac{1}{B}\sum_{b=1}^{B} S_b(\boldsymbol{\lambda}),$$

and the regularisation parameters are selected as $\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}\in\Lambda_\sigma\times\Lambda_\ell} S(\boldsymbol{\lambda})$. The final hyper-parameters $\boldsymbol{\theta}_{1:T}$ are obtained by solving (5) on the full dataset with the selected $\hat{\boldsymbol{\lambda}}$. For this work the selection was achieved using grid search, requiring $|\Lambda_\sigma|\,|\Lambda_\ell|\,B$ MAP optimisations. Each optimisation involves $O(T)$ evaluations of REML at cubic cost in the per-task sample size n, yielding overall complexity $O(|\Lambda_\sigma|\,|\Lambda_\ell|\,B\,T\,n^3)$, with the number of quasi-Newton iterations an implicit constant. For typical applications of BBPN, the number T of scalar quantities of interest can be large (e.g. $T \sim 10^3 - 10^5$), while the number n of simulator fidelities is typically small (e.g. $n \sim 10^1$). A linear complexity in T is therefore advantageous, while a cubic complexity in n is not problematic in this context.

4 Empirical Assessment

The aim of this section is to empirically compare the `independent`, `global`, and `sequential` approaches to BBPN. As baselines we also implemented `MOGP` and `SLFM` from Section 1.2, in each case deployed *within* the BBPN framework: these models retain the polynomial prior mean (2) and serve only to replace the covariance structure over $[0,\infty)^d \times \mathcal{T}$, with predictions still obtained by conditioning at $\mathbf{x} = \mathbf{0}$. Full details are contained in Section B.1. Our motivation is complex numerical simulations where achieving high accuracy is impractical, but to conduct this assessment we necessarily need to consider simpler simulations for which a high-accuracy ground truth can be computed.

To this end, two toy examples are considered. The first is the Lorenz system (Section 4.2); a nonlinear dynamical system which captures some of the features of more complex dynamical simulations. The second is eigenvalue computation via the QR algorithm (Section 4.3); a problem from numerical linear algebra for which bespoke PN methods have not been developed.

In both cases, the aim is to probabilistically predict the continuum quantities of interest from finite-resolution numerical output. Existing work has established theoretical and empirical accuracy of point estimators (Teymur et al., 2021; Oates et al., 2024; Blinded, 2026); our focus is on whether probabilistic predictions are *calibrated*. This requires quantitative measures of calibration, discussed next.

4.1 Assessing Predictive Calibration

Let $y_t = f(\mathbf{0}, t)$ denote a true (scalar) quantity of interest that we are attempting to predict, and let

$$\mu_t = \mathbb{E}[f(\mathbf{0},t)|\mathcal{D}_t], \qquad \sigma_t^2 = \mathbb{V}[f(\mathbf{0},t)|\mathcal{D}_t]$$

denote the predictive mean and variance from SPRE, each for $t \in \mathcal{T}$. From these we can obtain a $1-\alpha$

credible interval

$$\mathrm{CI}_{1-\alpha}(\mu_t, \sigma_t) = \big(\mu_t - z_{1-\alpha/2}\,\sigma_t,\ \mu_t + z_{1-\alpha/2}\,\sigma_t\big)$$

where $z_{1-\alpha/2} = \Phi^{-1}(1-\alpha/2)$ is the $(1-\alpha/2)$-quantile of $\mathcal{N}(0,1)$. Such predictions are said to be *well-calibrated* if their nominal credible intervals achieve the expected frequentist coverage:

$$\mathbb{P}(y_t \in \mathrm{CI}_{1-\alpha}(\mu_t, \sigma_t)) = 1-\alpha, \qquad \forall\, \alpha \in (0,1),$$

where here the probability arises from sampling an index t uniformly at random from $\mathcal{T}$. In practice it is too much to expect BBPN to be well-calibrated, so quantitative measures of miscalibration are needed.

Coverage The *actual coverage* at nominal level $1-\alpha$ is defined as

$$\hat{c}_{1-\alpha} = \frac{1}{T}\sum_{t=1}^{T} 1[y_t \in \mathrm{CI}_{1-\alpha}(\mu_t, \sigma_t)]\,,$$

and should equal the *notional coverage* $1-\alpha$ if predictions are well-calibrated.

Probability Integral Transform The *probability integral transform* (PIT) values $u_t = \Phi(z_t)$, where $z_t = (y_t - \mu_t)/\sigma_t$, should be approximately uniform over $[0,1]$ when predictions are well-calibrated. Departures from uniformity can be quantified using the Kolmogorov–Smirnov statistic

$$D_T = \sup_{u\in[0,1]} \big|\hat{F}_T(u) - u\big|,$$

where $\hat{F}_T$ is the empirical CDF of $\{u_t\}_{t=1}^T$.

4.2 Lorenz System

Our first experiment is the Lorenz system (Lorenz, 1963) which we previewed in Figure 1. This is a coupled system of ordinary differential equations (ODEs)

$$\dot{\mathbf{u}}(s) = \mathbf{F}(\mathbf{u}(s)), \qquad \mathbf{u}(0) = \mathbf{u}_0,$$

where

$$\mathbf{F}(\mathbf{u}) = \big(\,\sigma(u_2 - u_1),\ u_1(\rho - u_3) - u_2,\ u_1u_2 - \beta u_3\,\big)^\top,$$

with standard parameter values $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$, and initial condition $\mathbf{u}_0 = (-5,\ 5,\ 20)^\top$. The quantities of interest are the solution values $\mathbf{u}(s_t)$ at $T = 40$ uniformly-spaced query times s_t spanning $[0.05, 1.00]$, indexed by $t \in \{1,\ldots,T\}$; each scalar component is treated as a separate extrapolation target and we use $\mathbf{f}(x,t)$ as shorthand.

Experimental Setup To mimic more challenging situations where numerical accuracy is limited, here we consider approximation using the forward Euler scheme,

$$\mathbf{u}_{k+1} = \mathbf{u}_k + x\,\mathbf{F}(\mathbf{u}_k),$$

where the time step is $x \in (0,\infty)$. The final output $\mathbf{f}(x,t)$ represents a numerical approximation to $\mathbf{u}(s_t)$ and is obtained by linearly interpolating between the closest grid points $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$, i.e.

$$\mathbf{f}(x,t) = \boldsymbol{u}_k + \frac{s_t - xk}{x}\,(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) \tag{7}$$

where $xk \le s_t < x(k+1)$. A high-accuracy reference solution—treated as ground truth—is computed with the classical fourth-order Runge–Kutta method with step size $x = 4\times10^{-5}$, with not-a-knot cubic splines used to extend the solution beyond the discrete time grid in place of (7).

Implementation of SPRE For this experiment we fixed the index set to be $A = \{0,1\}$, based on the first order accuracy of the forward Euler method.

Each task was extrapolated from $n = 12$ simulator resolutions, corresponding to the step-size grid $x \in x_{\text{base}} \times \{0.4, 0.5, \ldots, 1.5\}$ with base step size $x_{\text{base}} = 0.01$. With $n = 12$ design points and $|A| = 2$, the matrix $\mathbf{V}_A^\top \mathbf{K}^{-1} \mathbf{V}_A$ remained well-conditioned in all experiments that we report.

Results The coverage and PIT histograms for all methods are reported in Figure 2; here we pooled results for all 3 coordinates of the Lorenz system to produce a single plot for each method (a break-down of these results per-coordinate is deferred to Table 1). It can be seen that `global` and `MOGP` led to under-confident predictive uncertainty in general (e.g. actual coverage 35% at nominal coverage 68% for `global`), while on the other hand `independent` and `SLFM` were over-confident; only `sequential` was well-calibrated. Point-estimation accuracy is also reported in Table 1, where the root mean squared error of the `sequential` posterior mean was smaller than that of all other methods considered. The under-confidence of `global` can be attributed to a small number of tasks requiring much larger predictive error-bars than the rest, typically at the later simulation times as seen in Figure 1. On the other hand, the over-confident predictions associated with `independent` are well-understood consequences of estimating hyper-parameters from a small dataset (cf. Karvonen et al., 2020). The actual estimated hyper-parameters are displayed in Figure 4, demonstrating that `sequential` avoids the substantial fluctuations between tasks that we see in `independent`, while also adapting to different tasks in a manner that `global` cannot. The mechanisms for under- and over-confidence, respectively, in `MOGP` and `SLFM` are more complicated, but this behaviour may be attributable to both hyper-parameter estimation from a finite dataset and the additional difficulty of modelling complex structured multivariate output with a 'generic' GP.

4.3 Eigenvalue Computation

Our second experiment focuses on eigenvalue computation, a task for which bespoke PN methods are yet

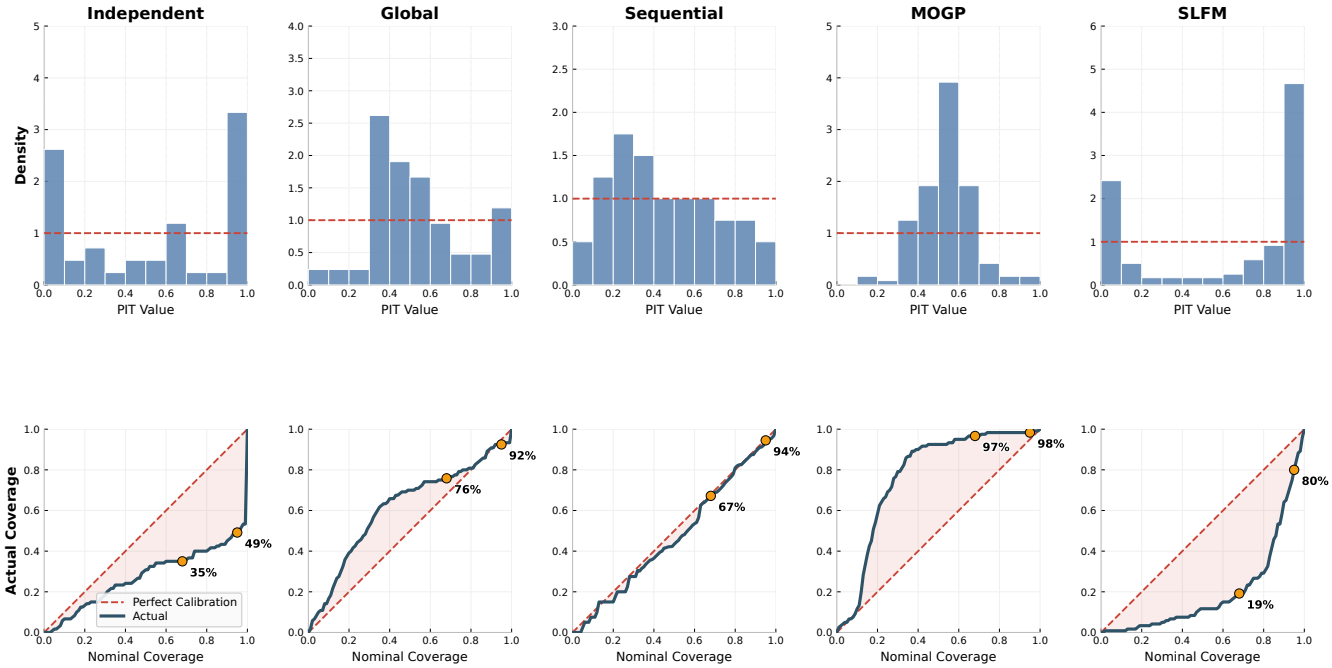


Figure 2: **Calibration diagnostics (Lorenz system Task)** Columns correspond to the methods `independent`, `global`, `sequential`, `MOGP` and `SLFM`. **Top row:** histograms of PIT values; the dashed horizontal line indicates the Uniform(0, 1) density. **Bottom row:** empirical coverage against nominal level; the dashed diagonal denotes perfect calibration. The actual coverage at 68% and 95% notional coverage are highlighted.

to be developed. In this example, the index t signifies the associated eigenvalue index, with $t = 1$ corresponding to the largest eigenvalue and $t = T$ to the smallest. The most standard algorithm in this context is the QR algorithm, which generates a sequence of diagonal approximations, denoted as $q_t(i)$, which converge to the eigenvalues λ_t of a symmetric matrix as the iteration count i increases. According to the foundational convergence theory (Wilkinson, 1965), the error decay is governed by

$$q_t(i) = \lambda_t + C_t \cdot r_t^i + O(r_t^{2i}), \qquad r_t = \left| \frac{\lambda_{t+1}}{\lambda_t} \right| < 1. \quad (8)$$

This formula implies that the convergence occurs exponentially as the number of iterations i is increased.

Parametrisation for SPRE SPRE requires a parameter $x \in (0, \infty)$ such that the numerical approximation becomes exact in the $x \to 0$ limit, and for which the numerical error admits a polynomial expansion, cf. (2). To reconcile the exponential convergence of the QR algorithm with the requirement of a polynomial expansion, we perform a coordinate transformation on the design variable:

$$x = e^{-c\,i}, \quad (9)$$

where the global decay rate c is empirically estimated. Substituting (9) into (8) yields:

$$q_t(i) = \lambda_t + C_t \cdot x^{c_t/c}, \qquad c_t = -\log r_t,$$

so that when c_t is close to c, the transformed response $q_t(i)$ converges approximately linearly with respect to x as $x \to 0$. To select a suitable value of c, we perform an ordinary least squares regression of $\log |\Delta q_t(i)|$ against i, utilizing the *median* slope across all eigenvalue indices to ensure robustness against slowly converging components (where $r_t \approx 1$). The original and transformed datasets are displayed in Figure 5.

Implementation of SPRE In the above parametrisation the convergence is approximately linear, and so for experiment we again fixed the index set to be $A = \{0, 1\}$.

Test Problem For empirical validation, we utilize a 2D discrete Poisson–Laplacian operator defined on an $l \times m$ grid, with $l = m = 10$ in our experiments, similarly to Teymur et al. (2021). The analytical eigenvalues for this system are given by

$$\lambda_{p,q} = 4 - 2\cos\left(\frac{p\pi}{l+1}\right) - 2\cos\left(\frac{q\pi}{m+1}\right),$$

where $1 \leq p \leq l$, and $1 \leq q \leq m$, so that we have access to the exact ground truth.

The (unshifted) QR algorithm was run for 20 iterations in total, with the fitting window restricted to iterates $i \in \{3, \ldots, 20\}$ (i.e. $n = 18$ data per task); the first two iterates were discarded as pre-asymptotic transients, since for the fastest-converging eigenvalues the initial iterates lie far from the asymptotic regime described by (8) and would otherwise corrupt the fitted polynomial trend.

Results Compared to the previous experiment, here there is greater heterogeneity across tasks; Figure 6 shows representative trajectories together with GP fits obtained using `independent`. Faster convergence of eigenvalue estimates leads to a pronounced linear structure in the transformed coordinate x, which is favourable

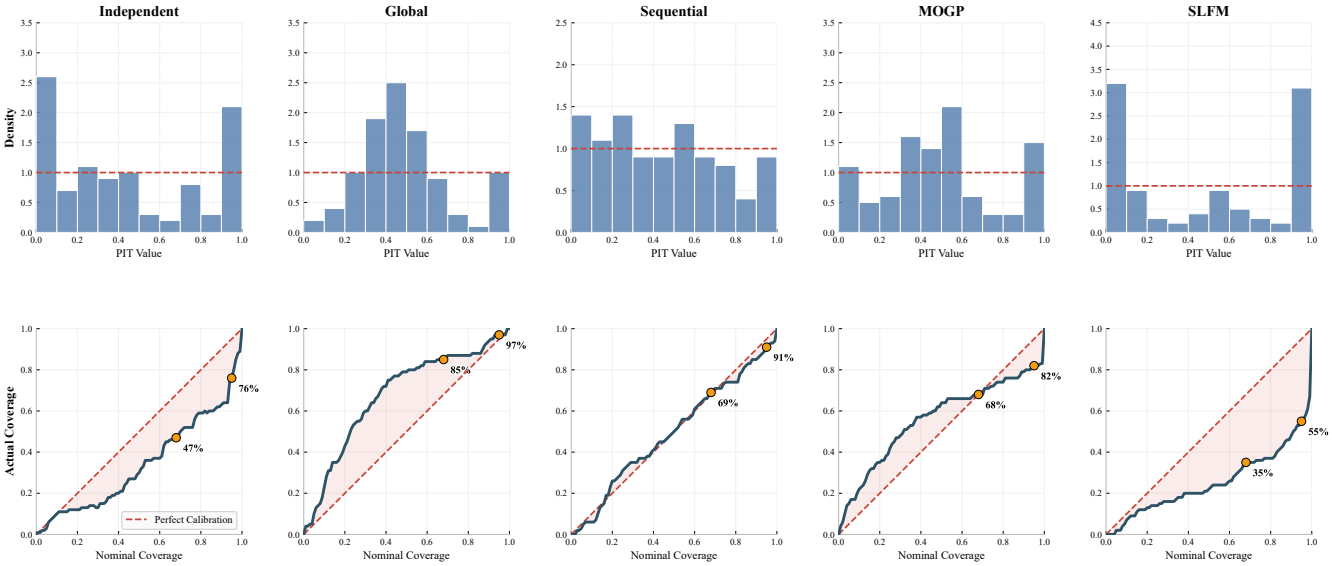


Figure 3: **Calibration diagnostics (QR Eigenvalue Task). Top row:** PIT histograms $u_i = \Phi(z_i)$ across all eigenvalues. **Bottom row (d–f):** Empirical coverage versus nominal level; orange markers indicate 68% and 95% reference levels. Sequential estimation closely follows the ideal diagonal compared to Independent and Global baselines. The actual coverage at 68% and 95% notional coverage are highlighted.

for being extrapolated, whereas slower convergence of eigenvalue estimates is more difficult to extrapolate from the dataset. The resulting differences in both curvature and uncertainty width across tasks indicate that `global` is inadequate, and that $\boldsymbol{\theta}_t$ should vary with the eigenvalue index $t \in \mathcal{T}$.

Figure 3 summarises predictive calibration across all T eigenvalue tasks. The `independent` and `SLFM` approaches produced pronounced U-shaped PIT distributions, with coverage falling short of nominal levels (e.g. actual 47% at the 68% level for `independent`), reflecting instability in per-task hyper-parameter estimation. In contrast, `global` yields a centrally concentrated PIT distribution, consistent with under-confidence: coverage remains above the diagonal at intermediate levels (actual 85% at nominal 68%), though it aligns reasonably well with the 95% target (actual 97%). The behaviour of `MOGP` was more complicated, exhibiting under-confidence at small and over-confidence at high nominal coverage. Our `sequential` method produces an approximately uniform PIT histogram and a calibration curve closely tracking the diagonal. Corresponding point-accuracy results are reported in Table 2; interestingly, `independent` very slightly out-performed `sequential` in root mean squared error (with both outperforming all other methods), but this small gain comes at the cost of considerable over-confidence in the associated predictive intervals.

The estimated hyper-parameters are displayed in Figure 7: as with the Lorenz experiment, `independent` estimation produces large spikes in σ_t and unstable ℓ_t, whereas `sequential` yields gradually varying hyper-parameters across the sequence of eigenvalues, underpinning its improved calibration.

Finally, we considered allowing the rate constant c_t to be t-dependent: Figure 8 presents GP fits for 15 eigenvalues covering a broad range of rates c_t, and these fits are broadly improved relative to Figure 6, suggesting this additional flexibility may be useful; we leave this direction for future work.

5 Discussion

Applications of BBPN have to-date been limited by a lack of effective approaches to handle multivariate output. We showed that a simple, computationally efficient `sequential` approach to hyper-parameter estimation can deliver reasonably well-calibrated probabilistic predictions for the limiting continuum quantities of interest. Our principal motivation was settings in which a realistic joint model for the continuum quantities of interest is infeasible, but our results suggested that a `sequential` strategy may also out-perform explicitly multivariate methods such as `MOGP` and `SLFM`. The principal limitation of the `sequential` approach is its reliance on an ordered index set $\mathcal{T}$ for which $f(\cdot, t)$ and $f(\cdot, s)$ are statistically similar when $t \approx s$; when this assumption fails—for example, when the ordering of $\mathcal{T}$ is arbitrary, or when the simulator output changes abruptly across the index—there is no reason to expect the regularised hyper-parameters to improve on `independent` estimation.

Several extensions merit investigation. The first-order Gaussian random walk prior (4) was adopted for its simplicity; richer alternatives, such as random walks with a full covariance matrix $\mathbf{Q}$ or vector auto-regressive models of lag greater than one, would enable more flexible modelling of the hyper-parameter trajectory, at the price of additional hyper-hyper-parameters to be estimated from limited data. Replacing the chain-structured prior with a Gaussian Markov random field on a general graph would broaden applicability to spatially indexed simulator output. Finally, allowing task-dependent convergence rates, as explored briefly in Section 4.3, may further improve calibration for problems exhibiting a wide range of convergence behaviour.

Acknowledgments

CJO was supported by EPSRC (EP/W019590/1) and a Philip Leverhulme Prize (PLP-2023-004).

References

M. A. Alvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

Blinded. Sparse probabilistic Richardson extrapolation. 2026. In preparation.

E. V. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. 2007.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5): 1190–1208, 1995.

C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

J. Cockayne, C. J. Oates, T. J. Sullivan, and M. Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019.

B. Da, Y.-S. Ong, A. Gupta, L. Feng, and H. Liu. Fast transfer Gaussian process regression with large-scale sources. *Knowledge-Based Systems*, 165:208–218, 2019.

O. Hamelijnck, T. Damoulas, K. Wang, and M. Girolami. Multi-resolution multi-task Gaussian processes. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.

D. A. Harville. Bayesian inference for variance components using only error contrasts. *Biometrika*, 61(2): 383–385, 1974.

D. Hegde and J. Cockayne. Learning to solve related linear systems. In *Proceedings of the 1st International Conference on Probabilistic Numerics*, pages 103–121, 2025.

P. Hennig, M. A. Osborne, and M. Girolami. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

T. Karvonen, G. Wynne, F. Tronarp, C. J. Oates, and S. Särkkä. Maximum likelihood estimation and uncertainty quantification for Gaussian process approximation of deterministic functions. *SIAM/ASA Journal on Uncertainty Quantification*, 8(3):926–958, 2020.

E. N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.

C. J. Oates, T. Karvonen, A. L. Teckentrup, M. Strocchi, and S. A. Niederer. Probabilistic Richardson extrapolation. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 2024.

B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60 (3):550–591, 2018.

C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA, 2006.

L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society A*, 210(459-470):307–357, 1911.

L. F. Richardson and J. A. Gaunt. The deferred approach to the limit. *Philosophical Transactions of the Royal Society A*, 226:223–361, 1927.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

O. Teymur, C. N. Foley, P. G. Breen, T. Karvonen, and C. J. Oates. Black box probabilistic numerics. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

H. Wackernagel. *Multivariate Geostatistics: An Introduction with Applications*. Springer Science & Business Media, 2003.

J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, 1965.

A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012.

X. Xi, F.-X. Briol, and M. Girolami. Bayesian quadrature for multiple related integrals. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

A Additional Results for the Lorenz System

This appendix presents the actual estimated hyper-parameters for the Lorenz system experiment, which are contained in Figure 4. Our methods were applied to each of the 3 coordinates of the Lorenz system in parallel, so that 3 sets of results are reported, each a different row in Figure 4. In addition we report detailed summary statistics for each coordinate in Table 1.

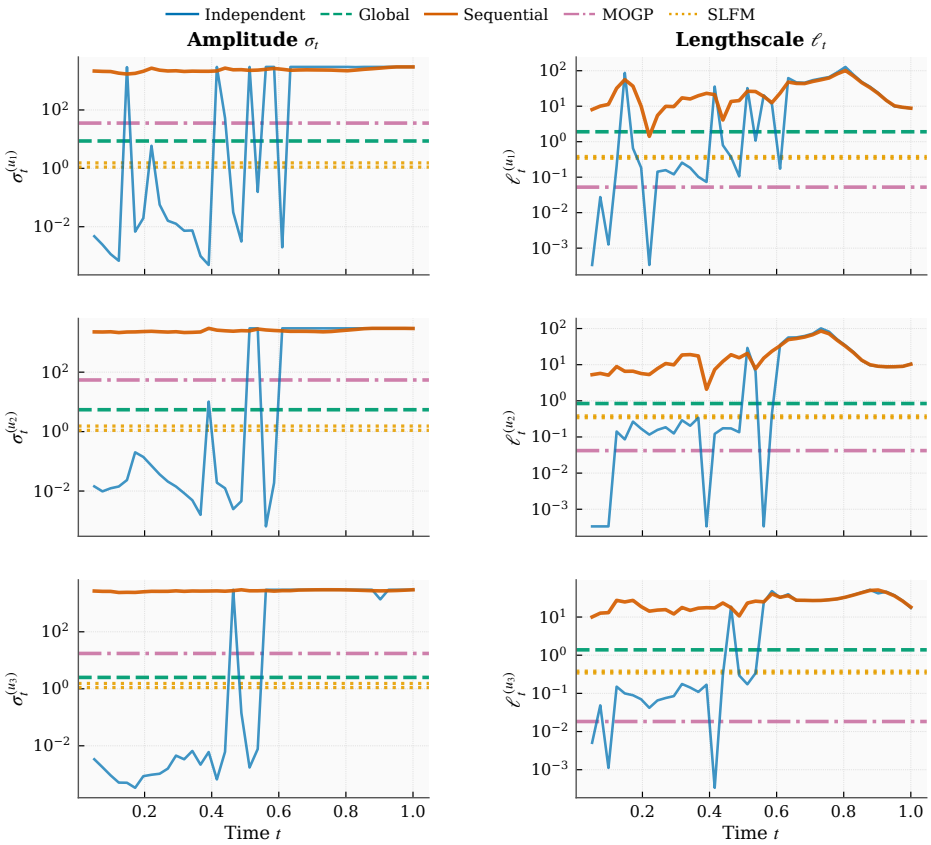


Figure 4: **Learned kernel hyper-parameter profiles across time for the Lorenz system.** Each row corresponds to a state component (x, y, z). Left column: amplitude $\sigma_t^{(\cdot)}$; right column: lengthscale $\ell_t^{(\cdot)}$.

Table 1: Predictive performance across coordinates and methods. **RMSE**: root mean squared error; **68%/95%**: empirical coverage at nominal levels 0.683 and 0.950; $\bar{z}/s_z$: mean and standard deviation of standardised residuals (well-calibrated $\Rightarrow \bar{z} \approx 0$, $s_z \approx 1$). Best RMSE per coordinate in **bold**.

Coord	Method	RMSE	68%	95%	$\bar{z}$	s_z
u_1	Independent	1.836×10^{-1}	0.325	0.650	+0.474	1.782
	Global	2.929×10^{-1}	0.825	0.875	+0.457	1.795
	Sequential	$\mathbf{1.822 \times 10^{-1}}$	0.775	0.975	−0.087	0.825
	MOGP	9.929×10^{-1}	0.950	0.975	+0.176	0.648
	SLFM	7.396×10^{-1}	0.175	0.825	+0.164	2.096
u_2	Independent	4.720×10^{-1}	0.300	0.575	+0.272	2.325
	Global	9.341×10^{-1}	0.775	0.850	+0.328	2.112
	Sequential	$\mathbf{4.652 \times 10^{-1}}$	0.725	0.925	+0.045	0.954
	MOGP	2.034×10^{0}	0.950	0.975	+0.110	0.428
	SLFM	1.587×10^{0}	0.225	0.750	+0.214	1.747
u_3	Independent	2.721×10^{-1}	0.300	0.600	+0.803	1.840
	Global	3.614×10^{-1}	0.900	1.000	+0.240	0.543
	Sequential	$\mathbf{2.643 \times 10^{-1}}$	0.725	0.900	+0.589	0.863
	MOGP	1.068×10^{0}	1.000	1.000	+0.081	0.175
	SLFM	7.953×10^{-1}	0.175	0.825	+0.710	1.480

B Additional Results for Eigenvalue Computation

This appendix presents additional empirical results for the eigenvalue computation task considered in Section 4.3. The original and transformed datasets are presented in Figure 5. The different nature of the extrapolation problems, indexed by t, together with the fits provided by `sequential`, is illustrated in Figure 6. The hyper-parameter values that were actually selected by each method are displayed in Figure 7. If we instead allow the rate constant c_t to be t-dependent, the fits provided by `sequential` can be improved; a representative selection of these GP fits is displayed in Figure 8. In addition we report detailed summary statistics in Table 2.

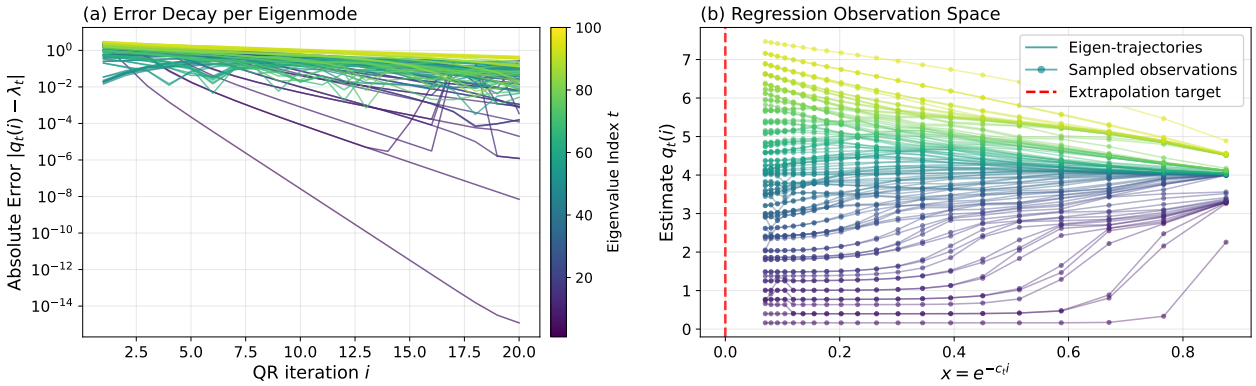


Figure 5: **High-dimensional spectral convergence and mapping characteristics ($l = 10, m = 10, n = 100$).** (a) The error decay trajectories for 100 eigenmodes illustrate the diversity of convergence rates within the spectrum. (b) Transformation of iterative estimates into the mapped regression space $x \in (0, 1]$. The clustering of trajectories demonstrates the structural consistency required for robust Gaussian Process extrapolation toward the theoretical limit $x \to 0$ (indicated by the dashed red line).

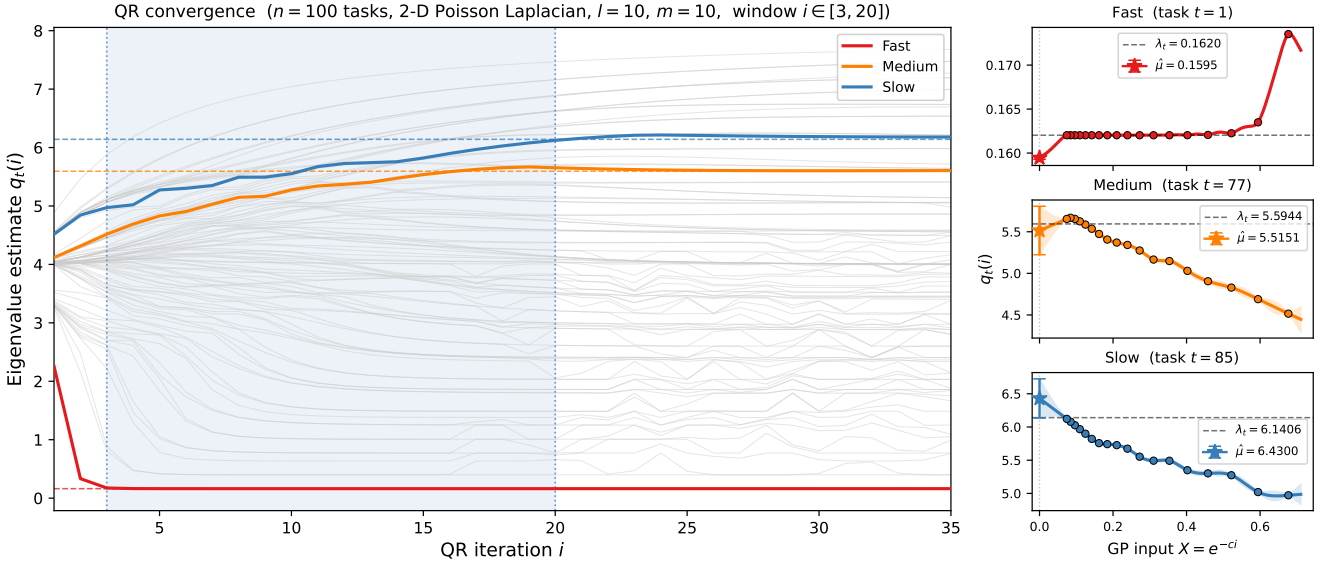


Figure 6: **Heterogeneous QR convergence and Gaussian process extrapolation. Left:** QR trajectories $q_t(i)$ for T eigenvalues, highlighting three representative tasks: Fast ($t = 1$), Medium ($t = 77$), and Slow ($t = 85$). The shaded region ($i \leq 20$) denotes the observation window used for training; dashed horizontal lines indicate the true eigenvalues λ_t. **Right:** GP posterior fits in the transformed coordinate $x = e^{-ci}$ for the highlighted tasks. Solid curves and shaded regions denote posterior means and 95% credible bands, respectively; circles denote observations. The dashed horizontal line marks the ground truth λ_t, and $\hat{\mu}$ is the extrapolated value at $x = 0$.

Figure 8 presents GP posterior fits for 15 selected eigenvalue tasks ($t = 4, \ldots, 18$) of the 2D Poisson–Laplacian, using per-task convergence rates c_t estimated individually from the observed QR iterates $\{q_t(i)\}$. Each panel displays the GP posterior mean (solid curve) and 95% credible interval (shaded band) as a function of the transformed input $x = e^{-c_t i}$, clipped to $x \leq 0.9$ to exclude early iterates dominated by transient behaviour. The dashed horizontal line marks the true eigenvalue λ_t, and the star marker at $x = 0$ denotes the extrapolated prediction together with its uncertainty. Panels are coloured by convergence rate c_t (plasma scale), with warmer colours indicating faster convergence. Panel (a) shows results from the `independent` method, in which each task optimises its own GP hyperparameters independently; Panel (b) shows results from the `sequential` method, in which hyperparameters are regularised across adjacent eigenvalue tasks via a geometric random walk prior. The `sequential` method yields noticeably tighter and more stable posterior intervals across tasks, particularly for those with irregular convergence behaviour (e.g. $t = 6, 7, 8, 15$), demonstrating the benefit of sharing statistical strength across the eigenvalue spectrum.

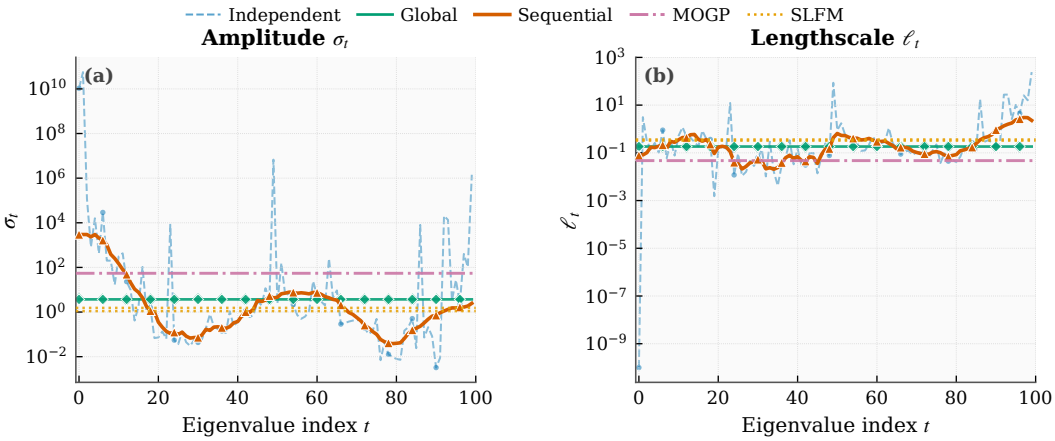
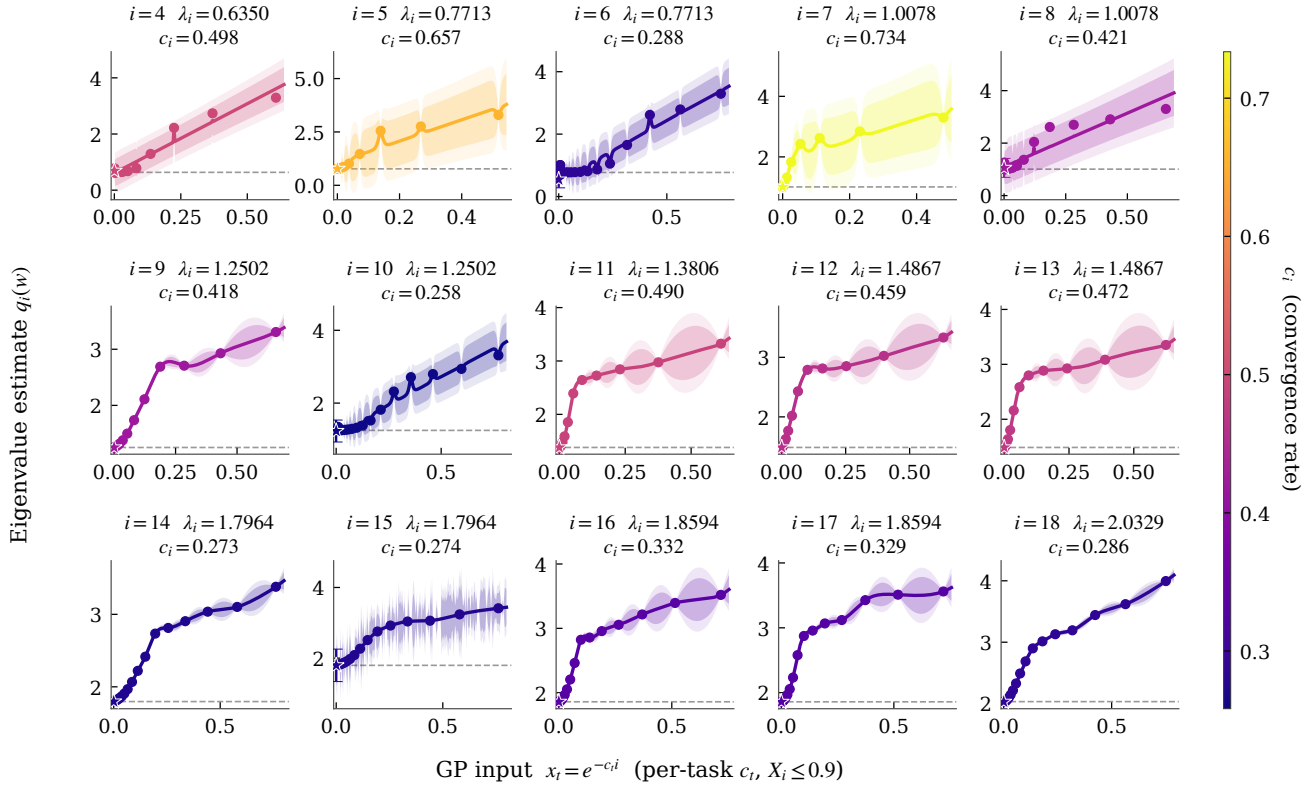


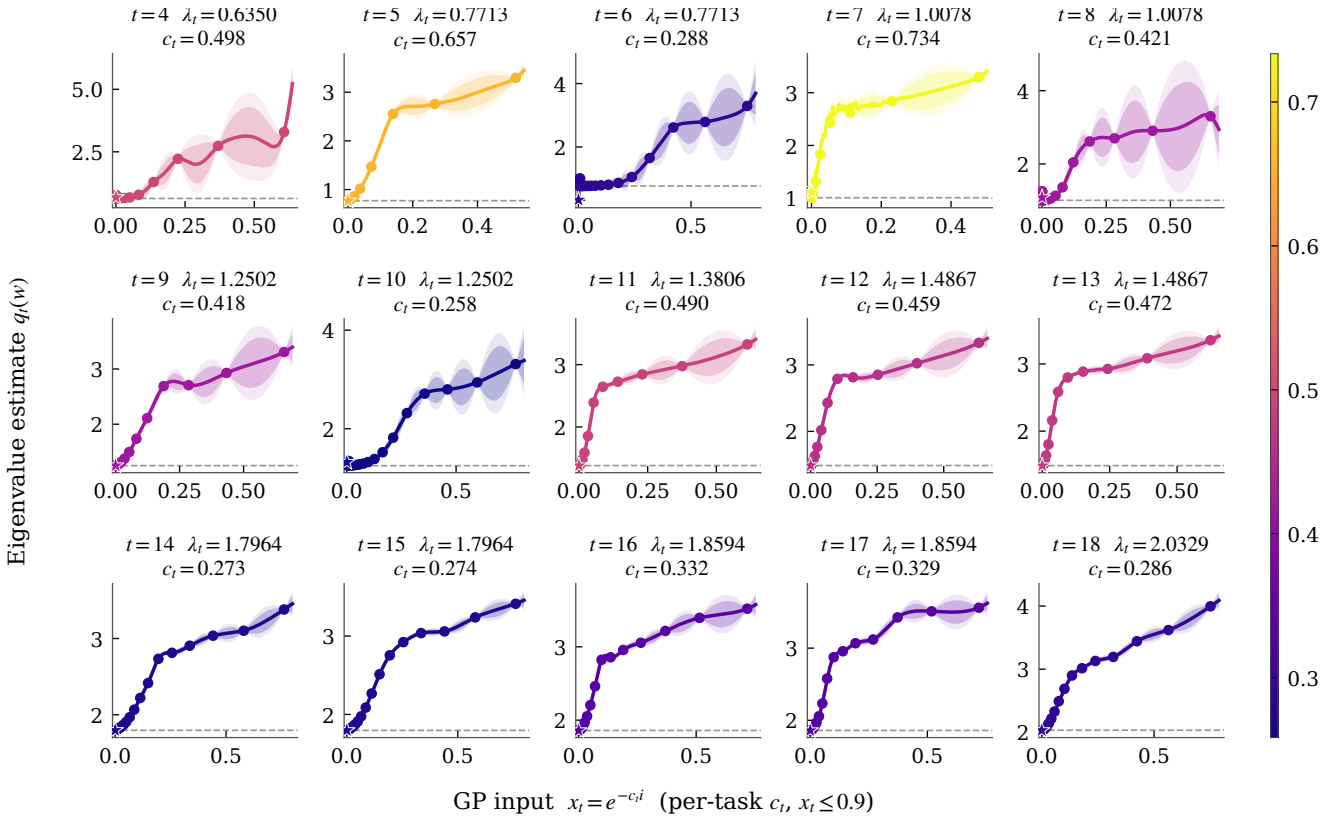
Figure 7: **Learned kernel hyperparameter profiles across eigenvalue tasks.** Each panel shows the estimated amplitude σ_i (left column) and lengthscale ℓ_i (right column) as a function of eigenvalue index i. The sequential model yields time-varying per-task hyperparameters, while MOGP and SLFM maintain fixed shared values across tasks.

Table 2: Predictive performance on the QR eigenvalue extrapolation problem. **RMSE**: root mean squared error; **68%/95%**: empirical coverage at nominal levels 0.683 and 0.950; $\bar{z}/s_z$: mean and standard deviation of standardised residuals (well-calibrated $\Rightarrow \bar{z} \approx 0$, $s_z \approx 1$). Best RMSE in **bold**.

Method	RMSE	68%	95%	$\bar{z}$	s_z
Independent	$\mathbf{1.502 \times 10^{-1}}$	0.480	0.760	-0.157	2.571
Global	2.433×10^{-1}	0.850	0.970	$+0.030$	0.754
Sequential	1.636×10^{-1}	0.700	0.910	-0.138	1.850
MOGP	2.264×10^{-1}	0.680	0.820	$+88.4$	837.3
SLFM	2.081×10^{-1}	0.350	0.550	-2.238	14.24



(a) Independent GP fits with per-task convergence rates c_t.



(b) Sequential GP fits with per-task convergence rates c_t.

Figure 8: GP posterior fits for 15 selected eigenvalue tasks using per-task convergence rates c_t, estimated individually from the observed QR iterates $\{q_t(i)\}$. Each panel shows the GP posterior mean (solid curve) and 95% credible interval (shaded band) as a function of $x = e^{-c_t i}$, together with the true eigenvalue λ_t (dashed horizontal line) and the extrapolated prediction at $x = 0$ (star marker). Panels are coloured by convergence rate c_t (plasma scale). **(a)** Independent method: each task optimises its own hyperparameters independently. **(b)** Sequential method: hyperparameters are regularised across adjacent tasks via a random walk prior, yielding tighter and more stable posterior intervals.

B.1 Implementation

For all experiments in the main text we employed the Matérn-$\frac{3}{2}$ kernel, for which the radial part is $\phi(r) = (1 + \sqrt{3}r)\exp(-\sqrt{3}r)$; (in)sensitivity to this choice is explored in Section C. The hyper-parameter candidate sets Λ_σ and Λ_ℓ were each of size 20, with values representing a logarithmic grid. A total of $B = 5$ folds of cross-validation were used. In addition we implement two 'generic' multivariate models as baselines, the details of which we now briefly report.

MOGP As an example of a multivariate model we consider `MOGP`. The prior mean was specified in the same manner as SPRE in (2), and a tensor product covariance function

$$\mathbb{C}[f(\mathbf{x}, t), f(\mathbf{x}', t')] = k_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}')k_{\mathcal{T}}(t, t'),$$

where the kernel $k_{\mathcal{T}}$ was parametrised as $k_{\mathcal{T}}(i, j) = [\mathbf{B}\mathbf{B}^\top + 10^{-6}\mathbf{I}]_{i,j}$ for a matrix $\mathbf{B}$ that represents an additional hyper-parameter to be estimated (this is known as the *intrinsic coregionalisation model*, due to Bonilla et al., 2007). The tensor product structure allows for a Kronecker decomposition of the full covariance matrix, reducing the complexity of marginal likelihood evaluations to $O(T^3 + n^3)$. For the experiments that we report, we jointly optimised the hyper-parameters $\boldsymbol{\theta}$ and $\mathbf{B}$ using L-BFGS-B over 5 random restarts, with a coregionalisation rank of 2 (i.e. the number of columns in $\mathbf{B}$).

SLFM As a second baseline we implemented `SLFM` in (1), with the latent factors g_i each modelled as an independent GP, with mean again specified in the same manner as SPRE in (2), and Matérn-$\frac{3}{2}$ covariance k_i, with individual hyper-parameters $\boldsymbol{\theta}_i$ to be estimated. Consequently, the covariance function has the form

$$\mathbb{C}[f(\mathbf{x}, t), f(\mathbf{x}', t')] = \sum_{i=1}^{p} c_i(t)c_i(t')k_{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}')$$

where the $c_i(t)$ are additional hyper-parameters to be estimated. For the experiments that we report, we jointly optimised the hyper-parameters $\boldsymbol{\theta}_i$ and $\mathbf{c}_i$ using L-BFGS-B over 5 random restarts. To control the difficulty of the hyper-parameter optimisation we fixed $p = 2$, so that `SLFM` encodes a relatively simple dependence structure over $\mathcal{T}$.

The code to reproduce our experiments is available in the `sequential-spre` repository on GitHub.

C Sensitivity to the Kernel

The aim of this appendix is to explore the (in)sensitivity of our conclusions to the choice of kernel k, which in the main text was Matérn-$\frac{3}{2}$. For this purpose we focused on the Lorenz experiment from Section 4.2. Figure 2 is reproduced using the Matérn-$\frac{5}{2}$ kernel in Figure 9. Figure 4 is reproduced using the Matérn-$\frac{5}{2}$ kernel in Figure 10. In both cases our conclusions are not substantially changed; the main difference is that `MOGP` now demonstrates a mixture of under-confidence (at low nominal coverage) and over-confidence (at high nominal coverage), whereas for the Matérn-$\frac{3}{2}$ kernel `MOGP` was over-confident at every nominal coverage level.

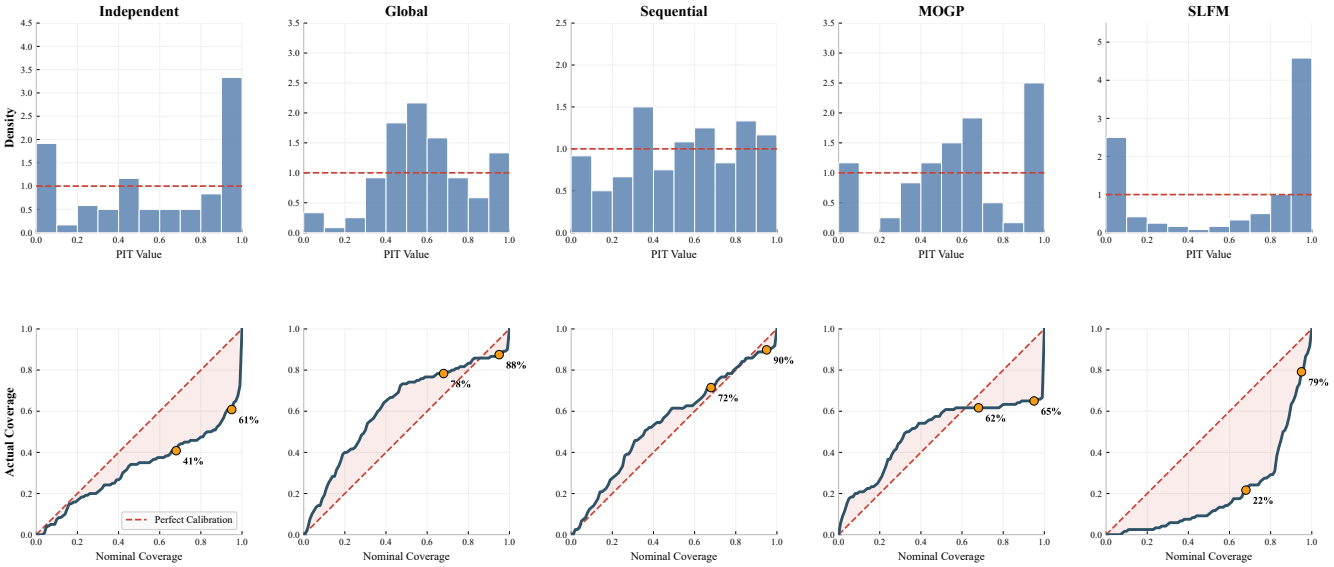


Figure 9: **Calibration diagnostics for the Lorenz system under an additional configuration using a Matérn-5/2 kernel.** Columns correspond to the methods `independent`, `global`, `sequential`, `MOGP`, and `SLFM`. **Top row:** histograms of PIT values, with the dashed horizontal line indicating the Uniform(0, 1) density. **Bottom row:** empirical coverage plotted against the nominal level, with the dashed diagonal representing perfect calibration. The results are consistent with those in Figure 2, with `sequential` providing the best-calibrated predictive intervals among the methods considered.

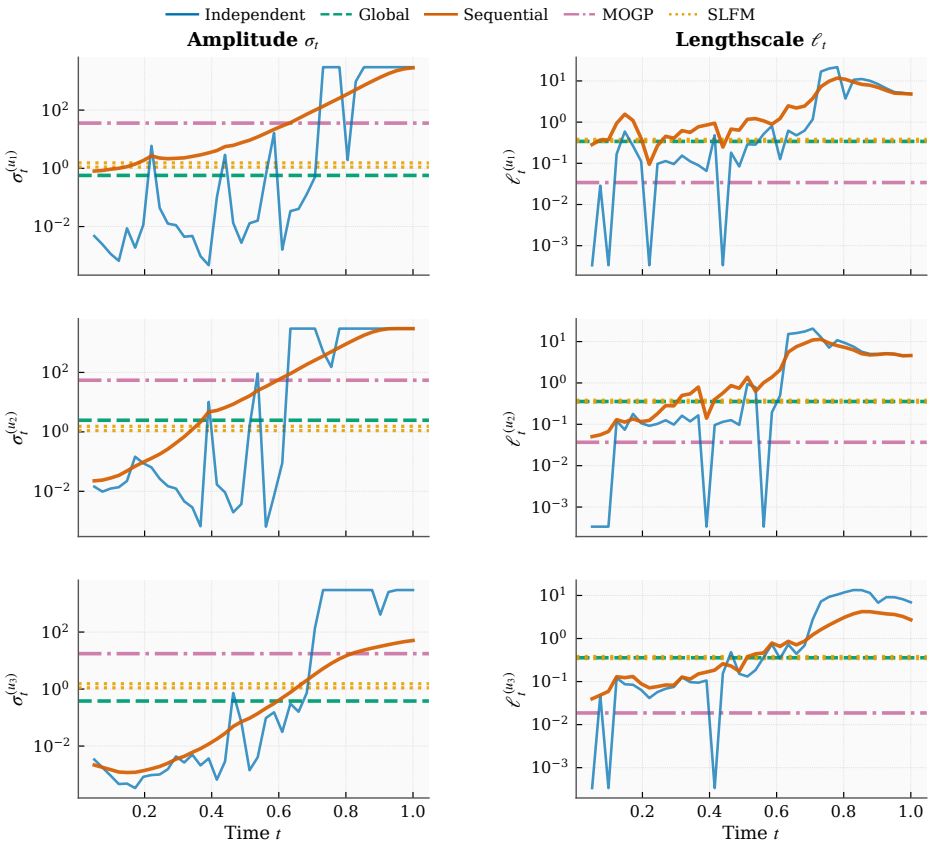


Figure 10: **Learned kernel hyperparameter profiles over time for the Lorenz system under an additional configuration using a Matérn-5/2 kernel.** Each row corresponds to one state component (x, y, z). The left column shows the amplitude $\sigma_t^{(\cdot)}$, and the right column shows the lengthscale $\ell_t^{(\cdot)}$. The `sequential` model yields smoothly varying hyperparameter trajectories over time, whereas the `independent` estimates exhibit erratic fluctuations, and the `global`, `MOGP`, and `SLFM` models retain constant shared values.