# Physics-Informed Variational State-Space Gaussian Processes

**Oliver Hamelijnck**
University of Warwick
oliver.hamelijnck@warwick.ac.uk

**Arno Solin**
Aalto University
arno.solin@aalto.fi

**Theodoros Damoulas**
University of Warwick
t.damoulas@warwick.ac.uk

## Abstract

Differential equations are important mechanistic models that are integral to many scientific and engineering applications. With the abundance of available data there has been a growing interest in data-driven physics-informed models. Gaussian processes (GPs) are particularly suited to this task as they can model complex, non-linear phenomena whilst incorporating prior knowledge and quantifying uncertainty. Current approaches have found some success but are limited as they either achieve poor computational scalings or focus only on the temporal setting. This work addresses these issues by introducing a variational spatio-temporal state-space GP that handles linear and non-linear physical constraints while achieving efficient linear-in-time computation costs. We demonstrate our methods in a range of synthetic and real-world settings and outperform the current state-of-the-art in both predictive and computational performance.

## 1 Introduction

Physical modelling is integral in modern science and engineering with applications from climate modelling [62] to options pricing [6]. Here, the key formalism to inject mechanistic physical knowledge are differential equations (DEs), which given initial and/or boundary values, are typically solved numerically [8]. In contrast machine learning is data-driven, and aims to learn latent functions from observations. However the increasing availability of data has spurred interest in combining these traditional mechanistic models with data-driven methods through physics-informed machine learning. These hybrids approaches aim to improve predictive accuracy, computational efficiency by leveraging both physical inductive biases with observations [30, 44].

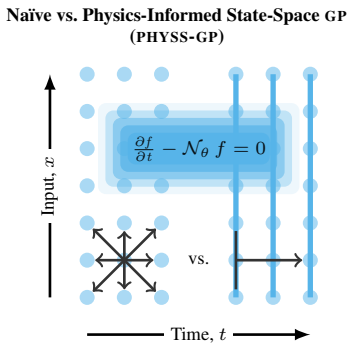


Figure 1: The state-space formalism allows for linear-time inference in the temporal dimension.

A principled way to incorporate prior physical knowledge is through Gaussian processes (GPs). GPs are stochastic processes and are a data-centric approach that facilitates the quantification of uncertainty. Recently AUTOIP was proposed in order to integrate non-linear physics into GPs [40], where solutions to ordinary and partial differential equations (ODEs, PDEs) are observed at a finite set of collocation points. This is an extension of the probabilistic meshless method (PMM, [12]) to the variational setting such that non linear equations can be incorporated. Similarly, [4] introduced HELMHOLTZ-GP, that constructs GP priors that adhere to curl and divergence-free constraints. Such properties are required for the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

successful modelling of electromagnetic fields [61] and ocean currents through the Helmholtz decomposition [4]. These approaches enable the incorporation of physics but incur a cubic computational complexity from needlessly computing full covariance matrices, as illustrated in Fig. 1. For ODEs (time-series setting), extended Kalman smoothers incorporate non-linear physics (EKS) [65, 34] and recover popular ODE solvers whilst achieving linear-in-time complexity through state-space GPs [58, 25].

In this work we propose a unified physics informed state-space GP (PHYSS-GP) that is a probabilistic models where mechanistic/physics knowledge is incorporated as an inductive bias. We can handle both linear and non-linear PDEs and ODEs whilst maintaining linear-in-time computational efficiency. We additionally derive a state-space variational inference algorithm that further reduces the computational cost in the spatial dimension. We recover EKS, PMM, and HELMHOLTZ-GP as special cases, and outperform AUTOIP in terms of computational efficiency and predictive performance. In summary:

1. We derive a state-space GP that can handle spatio-temporal derivatives with a computational complexity that is linear in the temporal dimension.
2. With this we derive a unifying state-space variational inference framework that allows the incorporation of both linear and non-linear PDEs whilst achieving a linear-in-time complexity and recovering state-of-the-art methods such as EKS, PMM and HELMHOLTZ-GP.
3. We further explore three approximations, namely a structured variational posterior, spatial sparsity, and spatial minibatching, that reduce the cubic spatial computational costs to linear.
4. We showcase our methods on a variety of synthetic and real-world experiments and outperform the current state-of-the-art methods AUTOIP and HELMHOLTZ-GP both in terms computational and predictive performance.

Code to reproduce experiments is available at `https://github.com/ohamelijnck/physs_gp`.

## 2 Background on Gaussian Processes

**Gaussian processes** A GP is a distribution on an infinite collection of random variables such that any finite subset is jointly Gaussian [50]. Given observations $\mathbf{X} \in \mathbb{R}^{N \times F}$ and $\mathbf{y} \in \mathbb{R}^N$ then

$$p(\mathbf{y}, f \,|\, \boldsymbol{\theta}) = \textstyle\prod_n^N p(y_n \,|\, f(\mathbf{x}_n), \boldsymbol{\theta})\, p(f \,|\, \boldsymbol{\theta}) \tag{1}$$

is a joint model where $p(f \,|\, \boldsymbol{\theta})$ is a zero mean GP prior with kernel $\mathbf{K}(\cdot,\cdot)$, $f(\mathbf{X}) \sim p(f(\mathbf{X}) \,|\, 0, \mathbf{K}(\mathbf{X},\mathbf{X}))$, and $\boldsymbol{\theta}$ are (hyper) parameters. We are primarily concerned with the spatio-temporal setting where we observe $N_\mathrm{t}$ temporal and $N_\mathrm{s}$ spatial observations $x_{\mathrm{t,s}} \in \mathbb{R}$, $y_{\mathrm{t,s}} \in \mathbb{R}$ on a spatio-temporal grid. Under a Gaussian likelihood, all quantities for inference and training are available analytically and, naïvely, carry a dominant computational cost of $\mathcal{O}((N_\mathrm{t}\, N_\mathrm{s})^3)$. For time series data, an efficient way to construct a GP over $f$ (and its time derivatives) is through the state-space representation of GPs. Given a Markov kernel, the temporal GP prior can be written as the solution of a discretised linear time-invariant stochastic differential equation (LTI-SDE), which at time $k$ is

$$\bar{\mathbf{f}}_{k+1} = \mathbf{A}\, \bar{\mathbf{f}}_k + q_k \quad \text{and} \quad y_k \,|\, \bar{\mathbf{f}}_k \sim p(y_k \,|\, \mathbf{H}\, \bar{\mathbf{f}}_k), \tag{2}$$

where $\mathbf{A}$ is a transition matrix, $q_k$ is Gaussian noise, $\mathbf{H}$ is an observation matrix, and $\bar{\mathbf{f}}$ is a $d$-dimensional vector of temporal derivatives $\bar{f} = [f(\cdot), \frac{\partial f(\cdot)}{\partial x}, \frac{\partial^2 f(\cdot)}{\partial x^2}, \cdots]^\top$. With appropriately designed states, matrices and densities, SDEs of this form represent a large class of GP models, and Kalman smoothing enables inference in $\mathcal{O}(N_\mathrm{t}\, d^3)$, see [56]. In the spatio-temporal setting, when the kernel matrix decomposes as a Kronecker product $\mathbf{K} = \mathbf{K}_t \otimes \mathbf{K}_s$, then with a Markov time kernel, a state space form is admitted. This takes a particularly convenient form where the state is $\bar{\mathbf{f}}_t = [\bar{f}((\mathbf{X}_s)_1, t), \cdots, \bar{f}((\mathbf{X}_s)_{Ns}, t)]^\top$, and inference requires $\mathcal{O}(N_\mathrm{t}(N_\mathrm{s}\, d)^3)$, see [60].

**Derivative Gaussian processes** One main appeal of GPs is that they are closed under linear operators. Let $\mathcal{D}\,[\cdot] = \mathcal{D}_\mathrm{t}\, \mathcal{D}_\mathrm{s}\,[\cdot]$ be linear functional that computes $D = d_t\, d_s$ space-time derivatives with $\mathcal{D}_\mathrm{t}\,[\cdot] = \left[\cdot, \frac{\partial \cdot}{\partial t}, \frac{\partial^2 \cdot}{\partial t^2}, \cdots\right]$ and $\mathcal{D}_\mathrm{s}\,[\cdot] = \left[\cdot, \frac{\partial \cdot}{\partial s}, \frac{\partial^2 \cdot}{\partial s^2}, \cdots\right]$, then at a finite set of index points, the joint prior between $\mathbf{f}$ and its time and spatial derivatives is

$$p(\bar{f}(\mathbf{X})) = \mathrm{N}\left(\, \mathcal{D}\,\mathbf{f} \;|\; \mathbf{0},\, \mathcal{D}\,\mathbf{K}(\mathbf{X},\mathbf{X})\,\mathcal{D}^*\,\right) \tag{3}$$

where $\bar{f}(\mathbf{X}) = \mathcal{D}\, f(\mathbf{X})$ and $\mathcal{D}^*$ is the adjoint of $\mathcal{D}$, meaning it operates on the second argument of the kernel [54]. When jointly modelling a single time and space derivative ($d_t = d_s = 1$) the latent functions are $\bar{\mathbf{f}} = [\mathbf{f}, \frac{\partial \mathbf{f}}{\partial \mathrm{s}}, \frac{\partial \mathbf{f}}{\partial \mathrm{t}}, \frac{\partial^2 \mathbf{f}}{\partial \mathrm{t}\, \partial \mathrm{s}}]^\top$ and the kernel is

$$\bar{\mathbf{K}} = \mathcal{D}\, \mathbf{K}(\mathbf{X}, \mathbf{X})\, \mathcal{D}^* = \begin{bmatrix} \mathbf{K} & - & - & - \\ \frac{\partial}{\partial s}\mathbf{K} & \frac{\partial}{\partial s}\mathbf{K}\frac{\partial}{\partial s}^\top & - & - \\ \frac{\partial}{\partial t}\mathbf{K} & \frac{\partial}{\partial t}\mathbf{K}\frac{\partial}{\partial s}^\top & \frac{\partial}{\partial t}\mathbf{K}\frac{\partial}{\partial t}^\top & - \\ \frac{\partial^2}{\partial t\, \partial s}\mathbf{K} & \frac{\partial^2}{\partial t\, \partial s}\mathbf{K}\frac{\partial}{\partial s}^\top & \frac{\partial^2}{\partial t\, \partial s}\mathbf{K}\frac{\partial}{\partial t}^\top & \frac{\partial^2}{\partial t\, \partial s}\mathbf{K}\frac{\partial^2}{\partial t\, \partial s}^\top \end{bmatrix}.$$

This is a multi-output prior whose samples are paths of $f$ with its corresponding derivatives. This prior is commonly known as a derivative GP and has found applications in monotonic GPs [51], input-dependent noise [41, 67] and explicitly modelling derivatives [59, 17, 43]. State-space GPs can be employed in the temporal setting since the underlying state computes $f(\mathbf{x})$ with its corresponding time derivatives. In Sec. 3.1, we extend this to the spatio-temporal setting.

## 3 Physics-Informed State-Space Gaussian Processes (PHYSS-GP)

We now propose a flexible generative model for incorporating information from both data observations and (non-linear) physical mechanics. We consider general non-linear evolution equations of the form

$$g(\mathcal{N}_\theta\, f) = \frac{\partial f}{\partial t} - \mathcal{N}_\theta\, f = 0 \tag{4}$$

with appropriate boundary conditions, where $f : \mathbb{R}^F \to \mathbb{R}$ is the latent quantity of interest and $\mathcal{N}_\theta$ is a non-linear differential operator [49]. We assume that $g : \mathbb{R}^{P \cdot D} \to \mathbb{R}$ is measurable, and is well-defined such that there are sensible solutions to the differential equation [25]. We wish to place a GP prior over $f$ and update our beliefs after 'observing' that it should follow the solution of the differential equation. In general this is intractable and can only be handled approximately. By viewing Eqn. (4) as a loss function that measures the residual between $\frac{\partial f}{\partial t}$ and the operator $\mathcal{N}_\theta\, f$ then the right hand side (0) are virtual observations. The PDE can now be observed at a finite set of locations known as collocation points. This is a soft constraint (*i.e.* $\mathbf{f}$ is not guaranteed to follow the differential equation), but it can handle non-linear and linear mechanisms. However, there are special cases, namely curl and divergence-free constraints, that can be solved exactly. This follows from properties of vectors fields, where $f$ defines a potential function where linear combinations of its partial derivatives define vector fields that enforce these properties. To handle both of these situations we propose the following generative model

$$\underbrace{\mathbf{F}_n = \mathbf{W} \cdot \left[\, \bar{f}_q(\mathbf{X}_n) \,\right]^\top}_{\text{Linear Mixing}}, \quad \underbrace{\bar{f}_q \sim \mathcal{GP}(\mathbf{0}, \bar{\mathbf{K}}_q)}_{\text{Independent GP Priors}}, \tag{5}$$

$$\underbrace{\mathbf{y}_n^{(\mathcal{O})} = \mathbf{H}_{\mathcal{O}}\, \mathbf{F}_n + \epsilon_{\mathcal{O}}}_{\text{Data}}, \quad \underbrace{\mathbf{0}_n^{(\mathcal{C})} = g(\mathbf{F}_n)}_{\text{Collocation Points}}, \quad \underbrace{\mathbf{y}_n^{(\mathcal{B})} = \mathbf{H}_{\mathcal{B}}\, \mathbf{F}_n + \epsilon_{\mathcal{B}}}_{\text{Boundary Values}}, \tag{6}$$

where $\bar{f}_q$ are derivative GPs (see Eqn. (3)) that are linearly mixed by $\mathbf{W} \in \mathbb{R}^{(\mathrm{P}\, D) \times (\mathrm{Q}\, D)}$, and $\mathbf{Y}^{(\mathcal{O})}, \mathbf{0}^{(\mathcal{C})} \in \mathbb{R}^{\mathrm{N} \times \mathrm{P}}$ are observations and collocation points over the P outputs and $\mathbf{Y}^{(\mathcal{B})} \in \mathbb{R}^{\mathrm{N} \times (\mathrm{P}\, D)}$ are boundary values over the derivatives of each output. The observation matrices $\mathbf{H}_{\mathcal{O}}, \mathbf{H}_{\mathcal{B}}$ simply select the relevant parts of $\mathbf{F}_n$. For further details on notation see App. A. In many case we want to observe the solution of the differential equation exactly, however in some cases it may be required to add observation noise $\epsilon_{\mathcal{C}}$ to the collocation points, whether for numerical reasons or to model inexact mechanics. This is a flexible generative model where different assumptions and approximations will lead to various physics informed methods such as AUTOIP, EKS, PMM, and HELMHOLTZ-GP that we will develop state space algorithms for. Additionally it is possible to learn missing physics by parameterising unknown terms in Eqn. (4) through the GP priors in Eqn. (6) (see App. B.2).

**Example 3.1** (EKS Prior and PMM)**.** We recover EKS style generative models (see Hennig et al. [25]) when the mixing weight is identity $\mathbf{W} = \mathbf{I}$, and $\epsilon_{\mathcal{C}}, \epsilon_{\mathcal{B}} \to 0$, and the non-linear transform $g$ is linearised. Let the prior be Markov $p(\bar{\mathbf{f}}) = \prod_k^{N_\mathrm{t}} p(\bar{\mathbf{f}}_k \,|\, \bar{\mathbf{f}}_{k-1})$ with marginals $p(\bar{\mathbf{f}}_k) = \mathrm{N}\left(\bar{\mathbf{f}}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-\right)$. By taking a first-order Taylor linearisation $g(\bar{\mathbf{f}}_k) \simeq g(\mathbf{m}_k^-) + \frac{\partial g(\mathbf{m}_k^-)}{\partial \mathbf{m}_k^-}\, \delta\bar{\mathbf{f}}_k$ with

$\delta\bar{\mathbf{f}}_k \sim \mathrm{N}\left(\mathbf{0},\, \mathbf{P}_k^-\right)$ the joint is

$$p(\begin{bmatrix}\bar{\mathbf{f}}_k \\ \mathbf{g}_k\end{bmatrix}) \simeq \mathrm{N}\left(\begin{bmatrix}\bar{\mathbf{f}}_k \\ \mathbf{g}_k\end{bmatrix} \;\middle|\; \begin{bmatrix}\mathbf{m}_k^- \\ g_k(\mathbf{m}_k^-)\end{bmatrix},\, \begin{bmatrix}\mathbf{I} \\ \frac{\partial g(\mathbf{m}_k^-)}{\partial \mathbf{m}_k^-}\end{bmatrix} \mathbf{P}_k^- \begin{bmatrix}\mathbf{I} \\ \frac{\partial g(\mathbf{m}_k^-)}{\partial \mathbf{m}_k^-}\end{bmatrix}^\top\right). \tag{7}$$

This is now a form that can directly be implemented into an extended Kalman smoothing algorithm [63]. When $\mathrm{Q} > 1$ the state $\bar{\mathbf{f}}$ is constructed by stacking the individual states of each latent [56]. With linear ODEs EKS coincides with PMM.

**Example 3.2** (HELMHOLTZ-GP and Curl and Divergence-Free Vector Fields in 2D)**.** Let $\mathbf{v} = [v_t, v_{s_1}, v_{s_2}]$ denote a 3D-vector field, then curl indicates the tendency of a vector field to rotate and divergence at a specific point indicates the tendency of the field to spread out. Curl and divergence-free fields follow

$$\nabla \times \mathbf{v} = 0 \;\; (\text{curl free}), \;\; \nabla \cdot \mathbf{v} \;\; = 0 \;\; (\text{div. free}) \tag{8}$$

where $\nabla = [\frac{\partial}{\partial \mathrm{t}}, \frac{\partial}{\partial \mathrm{s}_1}, \frac{\partial}{\partial \mathrm{s}_2}]$. Two basic properties of vector fields state that the divergence of a curl field and the curl of a derivative field are zero [3]. Let $[f_1, f_2]$ be scalar potential functions then

$$\mathbf{v}_{\text{curl}} = \nabla f_1 \;\; (\text{curl free}), \;\; \mathbf{v}_{\text{div}} = \nabla \times \nabla\, f_2 \;\; (\text{div. free}) \tag{9}$$

define curl and divergence-free fields. In 2D this simplifies to using the *grad* and *rot* operators over $\mathbf{v} = [v_{s_1}, v_{s_2}]$ (see [4]). Placing GP priors over $f_q$ we incorporate this into Eqn. (6) by defining

$$\mathbf{W}_{\text{grad}} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \mathbf{H}, \;\; \mathbf{W}_{\text{rot}} = \begin{bmatrix}0 & 1 \\ -1 & 0\end{bmatrix} \mathbf{H} \;\text{ where }\; \mathbf{H} \;\text{ selects }\; \left[\frac{\partial f}{\partial s_1}, \frac{\partial f}{\partial s_2}\right]. \tag{10}$$

HELMHOLTZ-GP is defined as the sum of GP priors over 2D curl and divergence-free fields [4].

### 3.1 A Spatio-Temporal State-Space Prior

The generative model in Eqn. (6) contains two complications: i) it includes potential non-linearities, and ii) the independent priors are defined over latent functions with their partial derivatives which substantially increases the computational complexity. We wish to tackle both issues through state-space algorithms that are linear-in-time. We begin by deriving a state-space model that observes derivatives across space and time (see App. A.3 for the simpler time-series setting). In Sec. 3.2 we further derive a state-space variational lower bound that will enable computational speeds up in the spatial dimension.

First, we show how Kronecker structure in the kernel allows us to rewrite the model as the solution to an LTI-SDE. From the definition of $\mathcal{D}$, the separable covariance matrix has a repetitive structure that can be represented through a Kronecker product. The gram matrix is

$$\mathcal{D}\, \mathbf{K}(\mathbf{x}, \mathbf{x})\, \mathcal{D}^* = \mathbf{K}_{\mathrm{t}}^{\mathcal{D}}(\mathbf{x}_{\mathrm{t}}, \mathbf{x}_{\mathrm{t}}) \;\otimes\; \mathbf{K}_{\mathrm{s}}^{\mathcal{D}}(\mathbf{x}_{\mathrm{s}}, \mathbf{x}_{\mathrm{s}}) \tag{11}$$

where $\mathbf{K}_{\cdot}^{\mathcal{D}\cdot} = \begin{bmatrix}\mathbf{K}_{\cdot\cdot} & \mathbf{K}_{\cdot\cdot}\tilde{\mathcal{D}}_{\cdot}^* \\ \tilde{\mathcal{D}}_{\cdot}\mathbf{K}_{\cdot\cdot} & \tilde{\mathcal{D}}_{\cdot}\mathbf{K}_{\cdot\cdot}\tilde{\mathcal{D}}_{\cdot}^*\end{bmatrix}$ and $\tilde{\mathcal{D}}_{\cdot}[\cdot] = (\mathcal{D}_{\cdot}[\cdot])_{1:}$ excludes the underlying latent function. To find a Kronecker form of the gram matrix over $\mathbf{X}$, we will exploit the fact that $\mathbf{X}$ is on a spatio-temporal grid and that the kernel is separable. Due to the separable structure a derivative over either the spatio (or temporal) dimension only affects the corresponding kernel, and so when considering $\mathbf{X}$, the gram matrix is still Kronecker structured:

$$\frac{\partial}{\partial s}\mathbf{K}(\mathbf{x}, \mathbf{x}) = \mathbf{K}_t(\mathbf{x}, \mathbf{x}) \cdot \frac{\partial}{\partial s}\mathbf{K}_s(\mathbf{x}, \mathbf{x}) \Rightarrow \frac{\partial}{\partial s}\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{K}_t(\mathbf{X}_t, \mathbf{X}_t) \otimes \frac{\partial}{\partial s}\mathbf{K}_s(\mathbf{X}_s, \mathbf{X}_s). \tag{12}$$

The full prior over (a permuted) $\mathbf{X}$ is now given as

$$p(\bar{f}(\mathbf{X})) \cong \mathrm{N}\left(\mathbf{0},\, \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_t, \mathbf{X}_t) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_s, \mathbf{X}_s)\right).$$

This is the form of a spatio-temporal Gaussian process with derivative kernels that can be immediately cast into a state-space form as in Eqn. (2) where $\mathbf{H} = \mathbf{I}$, as we want to observe the whole state, not just $\mathbf{f}$. The marginal likelihood and the GP posterior can now be computed using standard Kalman filtering and smoothing algorithms with a computational time of $O(N_{\mathrm{t}} \cdot (N_{\mathrm{s}} \cdot d_s \cdot d)^3)$. Inference in PHYSS-GP now follows Ex. 3.1 by recognising that the filtering state consists of the spatial points with there spatio-temporal derivatives. The EKS prior in Ex. 3.1 can now be simply extended to the PDE setting by placing colocation points on a spatio-temporal grid [35].

### 3.2 A State-Space Variational Lower Bound (PHYSS-VGP and PHYSS-EKS)

We now derive a variational lower bound for PHYSS-GP that maintains the computational benefits of state-space GPs. This acts as an alternative way of handling the non-linearity of $g$ in Eqn. (6), and will also enable the reduction of the cubic spatial computation complexity in Sec. 4. We start by focusing on the single latent function setting ($\mathrm{Q} = 1$) and collect all terms that relate to observations in Eqn. (6) with $p(\mathbf{Y} \,|\, \bar{\mathbf{f}}) = \prod_n^N p(\mathbf{y}_n^{(\mathcal{O})} | \mathbf{H}_{\mathcal{O}}\, \mathbf{F}_n)\, p(\mathbf{0}_n^{(\mathcal{C})} | g(\mathbf{F}_n))\, p(\mathbf{y}_n^{(\mathcal{B})} | \mathbf{H}_{\mathcal{B}}\, \mathbf{F}_n)$. VI frames inference as the minimisation of the Kullback–Leibler divergence between the true posterior and an approximate posterior, which leads the optimisation of the ELBO [28]:

$$\underset{q(\bar{\mathbf{f}} \,|\, \boldsymbol{\xi})}{\arg\max}\ \mathcal{L} = \mathbb{E}_{q(\bar{\mathbf{f}})} \left[ \log \frac{p(\mathbf{Y} \,|\, \bar{\mathbf{f}})\, p(\bar{\mathbf{f}})}{q(\bar{\mathbf{f}})} \right] \tag{13}$$

where we define the approximate posterior $q(\mathbf{f} \,|\, \boldsymbol{\xi}) \triangleq \mathrm{N}(\,\mathbf{f} \mid \mathbf{m},\, \mathbf{S}\,)$ as a free-form Gaussian with $\boldsymbol{\xi} = (\mathbf{m}, \mathbf{S})$ and $\mathbf{m} \in \mathbb{R}^{D\,N \times 1}$, $\mathbf{S} \in \mathbb{R}^{D\,N \times D\,N}$. The aim is to represent the approximate posterior as a state-space GP posterior, which will enable efficient computation of the whole evidence lower bound (ELBO). We will achieve this through the use of natural gradients. The natural gradient preconditions the standard gradient with the inverse Fisher matrix, meaning the information geometry of the parameter space is taken into account, leading to faster convergence and superior performance [2, 31, 27]. For Gaussian approximate posteriors the natural gradient has a simple form [26]

$$\boldsymbol{\lambda}_k = \lambda_{k-1} + \beta\, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = (1 - \beta)\, \widetilde{\boldsymbol{\lambda}}_{k-1} + \beta\, \frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}_k} + \boldsymbol{\eta} = \widetilde{\boldsymbol{\lambda}} + \boldsymbol{\eta} \tag{14}$$

where $\boldsymbol{\lambda} = (\mathbf{S}^{-1}\mathbf{m},\, {}^1\!/_2\, \mathbf{S}^{-1})$ and $\boldsymbol{\mu} = (\mathbf{m},\, \mathbf{m}\,\mathbf{m}^\top + \mathbf{S})$ are the natural and expectation parameterisations. This is known as conjugate variational inference (CVI) as $\widetilde{\boldsymbol{\lambda}}$ represent the natural parameters for the conjugate prior $\boldsymbol{\eta}$ [31, 10, 20, 72]. For now, we will assume that the likelihood is conjugate to ensure that $[\boldsymbol{\lambda}_k]_2$ is *p.s.d*, this will be relaxed in Sec. 5. The derivative of the ELL is

$$\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}]_2} = \sum_{\mathrm{t,s}}^{N_\mathrm{t}, N_\mathrm{s}} \frac{\partial}{\partial [\boldsymbol{\mu}]_2}\, \mathbb{E}_q \left[ \log p(\mathbf{Y}_{(\mathrm{t,s})} \,|\, \bar{\mathbf{f}}_{(\mathrm{t,s})}) \right], \tag{15}$$

where the expectation is under $q(\bar{\mathbf{f}}_{(\mathrm{t,s})})$, a $D$ dimensional Gaussian over the spatio-temporal derivatives at location $\mathbf{x}_{\mathrm{t,s}}$. Within the sum, the only elements of $[\boldsymbol{\mu}]_2$ whose gradient will propagate through the expectation are the $D \times D$ elements corresponding to these locations. These points are unique and so $\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}]_2}$ has some (permutated) block-diagonal structure, hence Eqn. (14) can be written as

$$q(\bar{\mathbf{f}}) \propto \prod_t^{Nt} \left[ \mathrm{N}(\widetilde{\mathbf{Y}}_t \,|\, \bar{\mathbf{f}}_t, \widetilde{\mathbf{V}}_t) \right]\, p(\bar{\mathbf{f}}) \tag{16}$$

where $\widetilde{\mathbf{Y}}_t$ is $D$-dimensional. The natural gradient update, *i.e.* $q(\bar{\mathbf{f}}_t)$ in moment parameterisation, can now be computed using Kalman smoothing in $\mathcal{O}(N_\mathrm{t} \cdot (N_\mathrm{s} \cdot d_s \cdot d)^3)$. Collecting $\widetilde{\mathbf{Y}} = \mathrm{vec}(\,[\widetilde{\mathbf{Y}}_t]\,), \widetilde{\mathbf{V}} = \mathrm{blkdiag}\left(\,[\widetilde{\mathbf{V}}_t]\,\right)$, then the ELBO can also be computed efficiently by substituting this form of $q(\bar{\mathbf{f}}_t)$ in

$$\mathcal{L} = \sum_{\mathrm{t,s}}^{N_\mathrm{t}, N_\mathrm{s}} \mathbb{E}_{q(\bar{\mathbf{f}}_{(\mathrm{t,s})})} \left[ \log p(\mathbf{Y}_{(\mathrm{t,s})} \,|\, \bar{\mathbf{f}}_{(\mathrm{t,s})}) \right] - \sum_t^{Nt} \mathbb{E}_{q(\bar{\mathbf{f}}_t)} \left[ \log \mathrm{N}(\widetilde{\mathbf{Y}}_t \,|\, \bar{\mathbf{f}}_t, \widetilde{\mathbf{V}}_t) \right] + \log p(\widetilde{\mathbf{Y}} \,|\, \widetilde{\mathbf{V}}) \tag{17}$$

where the first two terms only depend on $q(\mathcal{D}\,\mathbf{f}_t)$ and the final term is simply a by-product of running the Kalman filter, leading to a dominant computational complexity of $\mathcal{O}(N \cdot (N_\mathrm{s} \cdot d_s \cdot d)^3)$. This cost is linear in the datapoints ($N$) because the expected log likelihood above decomposes across all spatio-temporal locations. In summary we have shown that natural gradient is equivalent updating a block-diagonal likelihood that decomposes across time; hence the approximate posterior is computable via Kalman smoothing algorithms. Extending to multiple latent functions ($\mathrm{Q} > 1$) we define a full Gaussian approximate posterior that captures all correlations between the latent functions $q(\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_\mathrm{Q}) \triangleq \mathrm{N}\left(\,\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_\mathrm{Q} \mid \mathbf{m},\, \mathbf{S}\,\right)$ where $\mathbf{m} \in \mathbb{R}^{(N \times Q) \times 1}, \mathbf{S} \in \mathbb{R}^{(N \times Q) \times (N \times Q)}$. All the observation models in Eqn. (6) decompose across data points, hence Eqn. (16) is still block-diagonal and decomposes across time, except now each component is of dimension $\mathrm{Q} \times N_\mathrm{t}$ as it encodes the correlations of spatial points and their spatio-temporal derivatives across the latent functions. We denote this model as PHYSS-VGP and PHYSS-EKS when using a EKS prior (see Ex. 3.1).

**Theorem 3.1.** *Let the approximate posterior be (full) Gaussian* $q(\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_{\mathrm{Q}}) \triangleq \mathrm{N}\left(\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_{\mathrm{Q}} \mid \mathbf{m}, \mathbf{S}\right)$ *where* $\mathbf{m} \in \mathbb{R}^{(N\times Q)\times 1}, \mathbf{S} \in \mathbb{R}^{(N\times Q)\times(N\times Q)}$. *When* $g$ *is linear a single natural gradient step with* $\beta = 1$ *recovers the optimal solution* $p(\bar{\mathbf{f}}_1, \cdots, \bar{\mathbf{f}}_{\mathrm{Q}} \mid \mathbf{Y})$.

We prove this in App. A.5.4. This result not only demonstrates the optimality of our proposed inference scheme in the linear Gaussian setting, but confirms that we recover batch models like PMM and HELMHOLTZ-GP, as well as EKS (see Ex. 3.1).

## 4 Reducing the Spatial Computational Complexity

We now propose three approaches that reduce the cubic computational complexity in the number of spatial derivatives and locations. The first augments the process with inducing points that alleviate cubic costs associated with $N_{\mathrm{s}}$. The second is a structured variational approximation that defines the approximate posterior only over the temporal prior and alleviates cubic costs associated with $d_s$. Finally, we introduce spatial mini-batching that alleviates linear $N_{\mathrm{s}}$ costs. When used in conjunction, the dominant computation cost is $\mathcal{O}\left(N_{\mathrm{t}} \cdot d_s \cdot (M_s \cdot d_t)^3\right)$. These approximations are not only useful for the state-space setting and can readily be applied to reduce the computational complexity for batch variational models (such as AUTOIP). See App. B.1 for more details.

**Spatio-Temporal Inducing Points (PHYSS-SVGP)** In this first approximation, denoted by PHYSS-SVGP, we augment the full prior $p(\bar{\mathbf{f}})$ with inducing points. By defining these inducing points on a spatio-temporal grid, we will show that we can still exploit Markov conjugate operations through natural gradients. Let $\bar{\mathbf{u}} = \mathcal{D}\,\mathbf{u} \in \mathbb{R}^{M\times D}$ be inducing points at locations $\mathbf{Z} \in \mathbb{R}^{M\times F}$. From the standard SVGP formulation [27], the ELBO is

$$\mathcal{L} = \mathbb{E}_{q(\bar{\mathbf{f}}, \bar{\mathbf{u}})}\left[\log \frac{p(\mathbf{Y} \mid \bar{\mathbf{f}})\, p(\bar{\mathbf{u}})}{q(\bar{\mathbf{u}})}\right] \tag{18}$$

where $q(\bar{\mathbf{f}}, \bar{\mathbf{u}}) \triangleq p(\bar{\mathbf{f}} \mid \bar{\mathbf{u}})\, q(\bar{\mathbf{u}})$. By defining the inducing points on a spatio-temporal grid at temporal locations $\mathbf{X}_{\mathrm{t}} \in \mathbb{R}_{\mathrm{t}}^N$ and spatial $\mathbf{Z}_{\mathrm{s}} \in \mathbb{R}^{M_s\times(F-1)}$ then the marginal $p(\bar{\mathbf{f}} \mid \bar{\mathbf{u}})$ is Gaussian with mean

$$\mu_{\mathbf{F}\mid\mathbf{U}} = \left[\mathbf{I} \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_{\mathrm{s}}, \mathbf{Z}_{\mathrm{s}})\left(\mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_{\mathrm{s}}, \mathbf{Z}_{\mathrm{s}})\right)^{-1}\right] \bar{\mathbf{u}} \tag{19}$$

and variance given in Eqn. (41). This Kronecker structure allows us to again 'decouple' space and time, leading to natural gradient updates with block size $M_s \times D$, reducing the computational complexity to $\mathcal{O}(N\,(M_s \cdot d_s \cdot d)^3)$. For full details, see App. A.5.1.

**Structured Variational Inference (PHYSS-SVGP$_{\mathrm{H}}$)** This second approximation, denoted as PHYSS-SVGP$_{\mathrm{H}}$, defines the inducing points *only* over the temporal derivatives. This is a useful approximation as it can drastically reduce the size of the filter state, making it more computationally and memory efficient. We begin by defining the joint prior as

$$p(\mathbf{F}, \mathcal{D}_{\mathrm{t}}\,\mathbf{f}) = p(\mathbf{F} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f})\, p(\mathcal{D}_{\mathrm{t}}\,\mathbf{f})$$

where $p(\mathbf{F} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f})$ is a Gaussian conditional with mean

$$\mathbb{E}\left[\mathbf{F} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f}\right] = \left[\mathbf{I} \otimes \widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_{\mathrm{s}}, \mathbf{X}_{\mathrm{s}})\, \mathbf{K}_s(\mathbf{Z}_s, \mathbf{Z}_s)^{-1}\right] \mathcal{D}_{\mathrm{t}}\,\mathbf{f}, \quad \text{with } \widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_{\mathrm{s}}, \mathbf{X}_{\mathrm{s}}) = \begin{bmatrix} \mathbf{K}_s(\mathbf{X}_s, \mathbf{Z}_s) \\ \mathcal{D}_{\mathrm{s}}\,\mathbf{K}_s(\mathbf{X}_s, \mathbf{Z}_s) \end{bmatrix}.$$

We then define a structured variational posterior

$$q(\bar{\mathbf{f}}, \mathcal{D}_{\mathrm{t}}\,\mathbf{f}) \triangleq p(\mathbf{F} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f})\, q(\mathcal{D}_{\mathrm{t}}\,\mathbf{f}).$$

Substituting this into the ELBO we see that all the terms with the prior spatial derivatives cancel

$$\mathbb{E}_q\left[\log \frac{p(\mathbf{Y} \mid \bar{\mathbf{f}})\, \cancel{p(\bar{\mathbf{f}} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f})}\, p(\mathcal{D}_{\mathrm{t}}\,\mathbf{f})}{\cancel{p(\mathbf{F} \mid \mathcal{D}_{\mathrm{t}}\,\mathbf{f})}\, q(\mathcal{D}_{\mathrm{t}}\,\mathbf{f})}\right]$$

Again, the marginal $q(\mathcal{D}_{\mathrm{t}}\,\mathbf{f})$ maintains Kronecker structure, enabling Markov conjugate operations, leading to a computational cost of $\mathcal{O}(N \cdot d_s \cdot (N_{\mathrm{s}} \cdot d)^3)$, see App. A.5.2. These variational approximations can simply be applied to non-state-space variational approximation, see App. B.1.

**Spatial Mini-Batching** A standard approach for handling big data is through mini-batching where the ELL is approximated using only a data subsample [27]. Directly appling mini-batching would be of little computation benefit because computation of the ELBO requires running a Kalman smoother that iterates through all time points. Instead, we mini-batch by subsampling $B_\mathrm{s}$ spatial points

$$\mathrm{ELL} \approx \sum_{\mathrm{t}}^{N_\mathrm{t}} \frac{N_\mathrm{s}}{B_\mathrm{s}} \sum_{i}^{B_\mathrm{s}} \mathbb{E}_q \left[ \log p(\mathbf{Y}_{\mathrm{t,s}} \mid \bar{\mathbf{f}}_{\mathrm{t},i}) \right] \tag{20}$$

where $i$ is uniformly sampled. We used in conjunction with PHYSS-SVGP and PHYSS-SVGP$_\mathrm{H}$, this results in dominant costs of $\mathcal{O}(N_\mathrm{t}\,(M_s \cdot d_s \cdot d)^3)$ and $\mathcal{O}\left(N_\mathrm{t} \cdot d_s \cdot (M_s \cdot d)^3\right)$ when $B_s \ll N_\mathrm{s}$.

## 5 Handling the PSD Constraint

As discussed in Sec. 3.2 when the differential equation is non-linear, the model is no longer conjugate and the resulting natural gradients are not guaranteed to result in *p.s.d* updates. This issue has received some attention in the literature [53, 64, 39], but these approaches do not maintain an efficient conjugate representation. One distinction is [72], which uses the Gauss-Newton approximation to maintain conjugate operations. We now extend this to support spatial inducing points and non-linear transformations. Due to space we focus on PHYSS-SVGP, but see App. A.5.3 for further details. The troublesome term for the natural gradient update in Eqn. (14) is the Jacobian of the ELL *w.r.t.* to the second expectation parameter; which is not guaranteed to be *p.s.d* unless the ELL is log convex [39]. Focusing at a single location $n = (t, s)$:

$$\frac{\partial \mathrm{ELL}_n}{\partial [\boldsymbol{\mu}_k]_2} = \frac{\partial}{\partial \mathbf{S}_u} \mathbb{E}_{q(\bar{\mathbf{u}}_t)} \left[ \mathbb{E}_{p(\bar{\mathbf{f}}_n \mid \bar{\mathbf{u}}_t)} \left[ \log p(\mathbf{Y}_n \mid \bar{\mathbf{f}}_n) \right] \right]$$

we apply the Bonnet's and Price's theorem [38] to bring the differential inside the expectation and make a Gauss-Newton [19] approximation ensuring that the Jacobian is *p.s.d*

$$\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}_t]_2} \approx \sum_{n,p}^{N} \mathbb{E}_{q(\bar{\mathbf{u}}_t)} \left[ \mathbf{J}_{n,p}^\top \mathbf{H}_{n,p} \mathbf{J}_{n,p} \right], \text{ where } \mathbf{J}_{n,p} = \frac{\partial g_n(\mu_n)}{\partial \bar{\mathbf{u}}_t}, \ \mathbf{H}_{n,p} = \frac{\mathrm{d}^2 \log p(\mathbf{Y}_n \mid g_n)}{\mathrm{d}^2 g_n}, \tag{21}$$

and $g_{n,p} = g(\bar{\mathbf{u}}_n)$ (Eqn. (4)) and $\mu_n$ is the mean of $p(\bar{\mathbf{f}}_n \mid \bar{\mathbf{u}}_t)$ (Eqn. (19)). When using spatial mini-batching Eqn. (21) is also subsampled.

## 6 Related Work

From the optimality of natural gradients, in the conjugate setting, we exactly recover batch GP based models such as [68, 29, 4]. Our inference scheme also applies to models that do not require derivative information i.e. in $d_t = d_s = 1$. As a special case, we recover [20], but we have extended the inference scheme to support spatial mini-batching, allowing big spatial datasets to be used. The linear weighting matrix can be used to define a linear model of coregionalisation and its variants [7, 77, 42, 66] and through appropriately designed functionals also non-linear variants [73].

In Álvarez et al. [76] GP priors over the solution of differential equations are obtained through a stochastic forcing term but they only consider situations where the Greens function is available. In [22, 23, 57, 33], efficient state-space algorithms are derived but are limited to the temporal setting only. Similarly, Heinonen et al. [24], learn a 'free-form ODE'. In the spatio-temporal setting Krämer et al. [35] and Duffin et al. [14] (which builds [18]) derive extended Kalman filter algorithms. Additionally there are approaches to constraining GPs by linear differential equations [37, 1, 5]. More generally than [4] in [21] GP priors over the solutions to linear PDEs with constant coefficients are derived.

Beyond GP based models, physics informed neural networks (PINNs) incorporate physics by constructing a loss function between the network and the differential equation at a finite set of collocation points [48]. This amounts to a highly complex optimisation problem [36] bringing difficulties for training [70, 71] and uncertainty quantification (UQ) [16]. Current approaches to quantifying uncertainty in PINNs are based on dropout [75] and conformal predictions [47]. In recent years UQ and deep learning has received much attention however is limited by its computational cost [45].

Table 1: Test performance on the simulated damped pendulum. Time is the total wall clock time in seconds.

| MODEL | WHITEN | $C$ | TIME | RMSE | NLPD |
|---|---|---|---|---|---|
| PHYSS-GP | – | 10 | 96.33 | 0.22 | −0.09 |
| | | 100 | 112.2 | 0.05 | −0.38 |
| | | 500 | 138.98 | 0.05 | −0.72 |
| | | 1000 | 144.29 | 0.06 | −0.79 |
| AUTOIP | × | 10 | 153.52 | 0.35 | 0.41 |
| | | 100 | 195.25 | 0.2 | −0.08 |
| | | 500 | 1011.88 | 0.33 | 0.32 |
| | | 1000 | 5134.13 | 0.36 | 0.41 |
| | ✓ | 10 | 164.58 | 0.16 | −0.30 |
| | | 100 | 208.81 | 0.05 | −0.41 |
| | | 500 | 1088.31 | 0.05 | −0.75 |
| | | 1000 | 5656.62 | 0.05 | −1.39 |

Table 2: Test performance on the magnetic field strength experiment. Results are computed *w.r.t.* to the first output. Time is the average epoch time in seconds.

| | TIME | | | R SQUARED | | |
|---|---|---|---|---|---|---|
| SPATIAL SIZE | 5 | 10 | 20 | 5 | 10 | 20 |
| HELMHOLTZ-GP | 0.21 | 0.46 | 2.37 | 0.23 | 0.97 | 0.97 |
| PHYSS-GP | 0.43 | 0.60 | 1.16 | 0.21 | 0.97 | 0.97 |
| PHYSS-SVGP | 0.44 | 0.44 | 0.31 | 0.23 | 0.96 | 0.97 |
| PHYSS-VGP$_{\text{H}}$ | 0.29 | 0.40 | 0.35 | 0.65 | 0.86 | 0.93 |
| PHYSS-SVGP$_{\text{H}}$ | 0.29 | 0.28 | 0.16 | 0.65 | 0.61 | 0.74 |

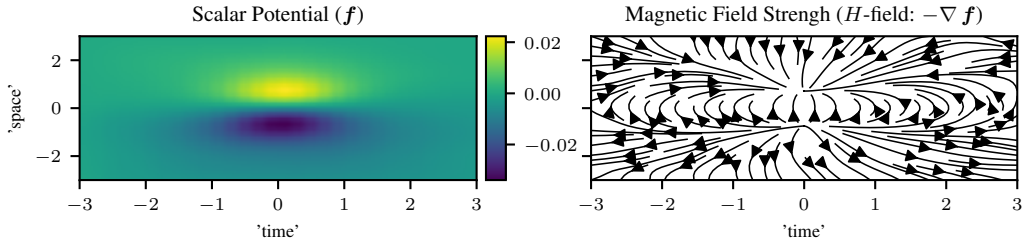


Figure 2: Curl free synthetic example. The left panel displays the learnt scalar potential functions by PHYSS-GP with $N_{\text{s}} = 20$, and the right panel illustrates the associated vector field.

## 7 Experiments

We now examine the performance of our PHYSS-GP methods on multiple synthetic and real-world datasets. We compare against a batch GP (no physical knowledge) and current state-of-art methods AUTOIP and HELMHOLTZ-GP. We provide more details on all experiments in App. B.

**Non-linear Damped Pendulum** In this first synthetic example, we consider learning the non-linear dynamics of a damped swinging pendulum. This is described by a second-order differential equation

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \sin(\,\theta\,) + b\,\frac{\mathrm{d}\theta}{\mathrm{d}t} = 0 \tag{22}$$

where $b > 0$. The first term is a non-linear forcing term, and the third is the damping term. With $b = 0.2$ we simulate a solution using Euler's method [9] and generate 20 points in $t \in [0, 6]$ for training and 200 in $t \in [6, 30]$ for testing, with additive Gaussian noise of variance $0.01$.

We are interested in *i)* the effect of the number of collocation points, *ii)* the effect of the optimisation algorithm. To answer these questions, we compare against AUTOIP on $[10, 100, 500, 1000]$ collocation points, with and without whitening. Results are tabulated in Table 1. As expected, the predictive RMSE of all models decreases as the number of collocation points increases. Due to the cubic complexity of AUTOIP, the total time significantly increases as the number of collocation points increases. For example, when using $1000$ collocation points, AUTOIP is $\approx 39$ times slower than PHYSS-GP. Interestingly, the un-whitened case performs poorly, possibly due to the nonlinearity of the differential equation making optimisation difficult. This indicates that either whitening or natural gradients are required to handle the non-linearity arising due to the differential equation.

**Curl-free Magnetic Field Strength** In this experiment, we consider modelling the magnetic field strength of a dipole $H(r) = -\nabla\psi(r)$, where $\psi(r) = \mathbf{m}\cdot\mathbf{r}/|\mathbf{r}|^3$ is a scalar potential function [11]. Labelling the input dimensions as 'time', 'space' and '$z$', we let $\mathbf{m} = [0, 1, 0]$ and generate observations from a spatio-temporal grid with $N_{\text{t}} = 50$, and $N_{\text{s}} = [5, 10, 20]$, at $z = 1$. $H(\mathbf{r})$ is a curl-free field and so we compare the curl free part of HELMHOLTZ-GP against PHYSS-GP and its variants. HELMHOLTZ-GP and PHYSS-GP are equivalent models (as this is the conjugate setting,

Table 3: Test performance on the diffusion-reaction system. Time is the total wall clock time in seconds. PHYSS-EKS significantly outperforms all models, and due to the EKS prior only requires a 1 epoch for inference. PHYSS-SVGP$_{\text{H}}$ achieves the same performance as AUTOIP but is over twice as fast.

| MODEL | RMSE | NLPD | CRPS | TIME | EPOCHS |
|---|---|---|---|---|---|
| PHYSS-EKS | **0.09** | **−1.26** | **0.038** | $\mathbf{1.1 \times 10^3}$ | 1 |
| PHYSS-SVGP | 0.19 | 6.80 | 0.093 | $1.4 \times 10^4$ | 20000 |
| PHYSS-SVGP$_{\text{H}}$ | 0.17 | 1.69 | 0.077 | $4.8 \times 10^3$ | 20000 |
| AUTOIP | 0.17 | −0.29 | 0.065 | $1.1 \times 10^4$ | 20000 |

Theorem 3.1), and recover the same posterior and predictive distribution (up to numerical precision). However, due to the cubic-in-time complexity HELMHOLTZ-GP, at larger spatial sizes, is over 2-times slower. The hierarchical approximation is substantially faster than PHYSS-GP and performs similarly. As expected when introducing sparsity both PHYSS-SVGP and PHYSS-SVGP$_{\text{H}}$ are even faster; however, this is compensated by a slight drop in predictive performance. See Fig. 2 and Table 2.

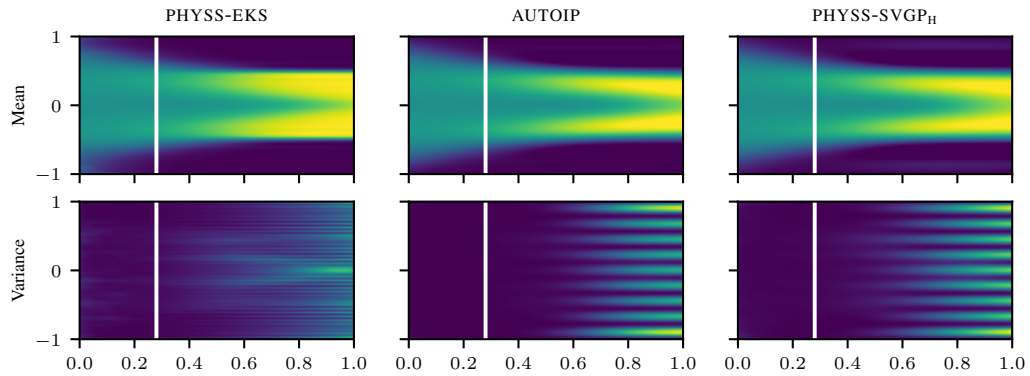


Figure 3: Results on the diffusion reaction system. The top row denotes the predictive mean, and the bottom the 95% confidence intervals. The white line denotes where the training data ends. Only PHYSS-EKS captures the sharp boundaries, due to the IWP kernel. PHYSS-SVGP$_{\text{H}}$ recovers a similar solution to AUTOIP but at half the computational cost.

**Diffusion-Reaction System** Consider a diffusion-reaction system given by an Allen-Cahn equation

$$\frac{\partial u}{\partial t} - 0.00001 \frac{\mathrm{d}^2 u}{\mathrm{d} x^2} + 5\,u^3 - 5\,u = 0 \tag{23}$$

where $x \in [-1, 1]$, $t \in [0, 1]$, $u(0, x) = x^2 \cos(\pi\, x)$, $u(t, -1) = u(t, 1)$ and $\frac{\partial u}{\partial x}(t, -1) = \frac{\partial u}{\partial x}(t, 1)$. Following [40], we use the solution provided by [49] and sample 256 training examples from $t \in [0, 0.28]$. We compare PHYSS-EKS (where $g$ is linerized in the EKS prior), PHYSS-SVGP and PHYSS-SVGP$_{\text{H}}$ against AUTOIP. Following [40], we use a learning rate of 0.001 for Adam. For AUTOIP, we place 100 collocation points across the whole input domain on a regular grid. For both PHYSS-SVGP, and PHYSS-SVGP$_{\text{H}}$ we require more collocation points in the temporal dimension and place them on a regular grid of size $20 \times 10$. For PHYSS-EKS we use an integrated Wiener kernel (IWP) on time [56] and place $100 \times 40$ collocation points. We are unable to place more collocation for AUTOIP due to computational limits. Results are presented in Fig. 3 and Table 3. PHYSS-EKS requires only a single epoch and can better handle the sharp boundaries. Our method PHYSS-SVGP$_{\text{H}}$ is over twice as fast as AUTOIP whilst achieving similar predictive RMSE.

**Ocean Currents** We now model oceanic currents in the Gulf of Mexico. We follow [4] and use the dataset provided by D'Asaro et al. [15] that has information from over $1,000$ buoys. We focus on the region in long. $[-90, -84.5]$, lat. $[26, 30]$ on 2016-02-25, by computing hourly averages. This results in $N = 42,243$ observations, and we construct a test-train split on 0.1 per cent of the data. It is infeasible to run HELMHOLTZ-GP due to data size (in Berlinghieri et al. [4], observations from only 19 buoys are used with $N = 55$). However, we run PHYSS-SVGP$_{\text{H}}$ with 50 spatial inducing

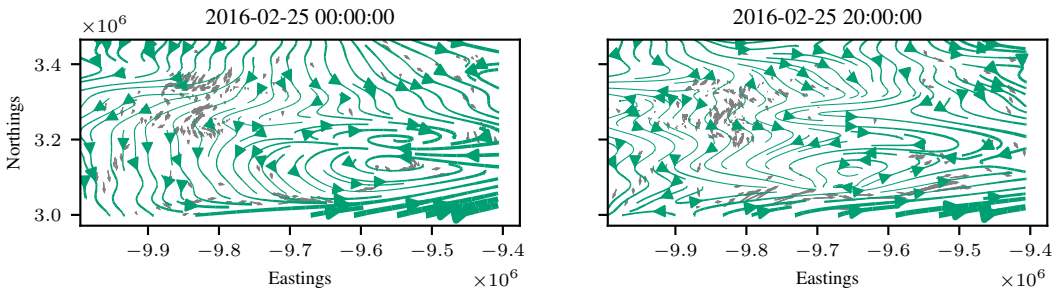


Figure 4: Predicted ocean currents by PHYSS-SVGP$_{\text{H}}$. True observations are in grey, and predictions in green. The thickness of the line represents uncertainty and is computed by the L2 norm of the standard deviations across both outputs.

points and a spatial mini-batch size of 10, and plot results in Fig. 4. Our predictions are in excellent agreement with the test data, achieving an RMSE of $0.14$, NLPD of $-0.52$, CRPS of $0.078$, and an average run-time of $1.86(s)$ per epoch.

## 8 Conclusion

We introduced a physics-informed state-space GP that integrates observational data with physical knowledge. Within the variational inference framework, we derived a computationally efficient algorithm that uses Kalman smoothing to achieve linear-in-time costs. To gain further computational speed-ups, we proposed three approximations with inducing points, spatial mini batching and structured variational posteriors. When used in conjunction, they allow us to handle large-scale spatiotemporal problems. The bottleneck is always the state size, where nearest neighbours GPs [13, 74] could be explored. For highly non-linear problems, future directions could explore deep approaches [52] or more flexible kernel families [69]. One limitation is the use of the collocation method which is only enforcing the differential equation point wise, whilst future work could look at the more general methods of weighted residuals [46].

## Acknowledgments and Disclosure of Funding

OH acknowledges funding from The Alan Turing Institute PhD fellowship programme and the UKRI Turing AI Fellowship (EP/V02678X/1). AS acknowledges support from the Research Council of Finland (339730). TD acknowledges support from UKRI Turing AI Acceleration Fellowship (EP/V02678X/1) and a Turing Impact Award from the Alan Turing Institute. The authors acknowledges the University of Warwick Research Technology Platform (aquifer) for assistance in the research described in this paper. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC-BY) license to any Author Accepted Manuscript version arising from this submission.

## References

[1] C. G. Albert. Gaussian processes for data fulfilling linear differential equations. In *Proceedings*, volume 33, page 5. MDPI, 2019.

[2] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[3] G. B. Arfken, H. J. Weber, and F. E. Harris. *Mathematical Methods for Physicists: A Comprehensive Guide*. Academic Press, 2011.

[4] R. Berlinghieri, B. L. Trippe, D. R. Burt, R. Giordano, K. Srinivasan, T. Özgökmen, J. Xia, and T. Broderick. Gaussian processes at the Helm (holtz): A more fluid model for ocean currents. *arXiv preprint arXiv:2302.10364*, 2023.

[5] A. Besginow and M. Lange-Hegermann. Constraining Gaussian processes to systems of linear ordinary differential equations. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 29386–29399. Curran Associates, Inc., 2022.

[6] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[7] E. V. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20 (NIPS)*. Curran Associates, Inc., 2008.

[8] D. Borthwick. *Introduction to Partial Differential Equations*. Universitext. Springer International Publishing, 2017.

[9] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2016.

[10] P. E. Chang, W. J. Wilkinson, M. E. Khan, and A. Solin. Fast variational learning in state-space Gaussian process models. In *30th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.

[11] T. Chow. *Introduction to Electromagnetic Theory: A Modern Perspective*. Jones and Bartlett Publishers, 2006.

[12] J. Cockayne, C. Oates, T. Sullivan, and M. Girolami. Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. *AIP Conference Proceedings*, 1853(1), 06 2017.

[13] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016.

[14] C. Duffin, E. Cripps, T. Stemler, and M. Girolami. Low-rank statistical finite elements for scalable model-data synthesis. *Journal of Computational Physics*, 463:111261, 2022.

[15] E. D'Asaro, C. Guigand, A. Haza, H. Huntley, G. Novelli, T. Özgökmen, and E. Ryan. Lagrangian submesoscale experiment (LASER) surface drifters, interpolated to 15-minute intervals, 2017. URL `https://data.gulfresearchinitiative.org/data/R4.x265.237:0001`.

[16] C. Edwards. Neural networks learn to speed up simulations. *Communications of the ACM*, 65: 27–29, 04 2022.

[17] D. Eriksson, K. Dong, E. Lee, D. Bindel, and A. G. Wilson. Scaling Gaussian process regression with derivatives. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*. Curran Associates, Inc., 2018.

[18] M. Girolami, E. Febrianto, G. Yin, and F. Cirak. The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions. *Computer Methods in Applied Mechanics and Engineering*, 375:113533, 2021.

[19] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.

[20] O. Hamelijnck, W. J. Wilkinson, N. A. Loppi, A. Solin, and T. Damoulas. Spatio-temporal variational Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021.

[21] M. Harkonen, M. Lange-Hegermann, and B. Raita. Gaussian process priors for systems of linear partial differential equations with constant coefficients. In *International Conference on Machine Learning*, pages 12587–12615. PMLR, 2023.

[22] J. Hartikainen and S. Särkkä. Sequential inference for latent force models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 311–318, Arlington, Virginia, USA, 2011. AUAI Press.

[23] J. Hartikainen, M. Seppänen, and S. Särkkä. State-space inference for non-linear latent force models with application to satellite orbit prediction. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 723–730. Omnipress, 2012.

[24] M. Heinonen, Çagatay Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki. Learning unknown ODE models with Gaussian processes. In *International Conference on Machine Learning (ICML)*, 2018.

[25] P. Hennig, M. Osborne, and H. Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

[26] J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the conjugate exponential family. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2888–2896. Curran Associates Inc., 2012.

[27] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290. AUAI Press, 2013.

[28] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14, 2013.

[29] C. Jidling, N. Wahlström, A. Wills, and T. B. Schön. Linearly constrained Gaussian processes. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*. Curran Associates, Inc., 2017.

[30] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[31] M. E. Khan and W. Lin. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXv preprint arXiv:1412.6980*, 2014.

[33] N. Krämer and P. Hennig. Linear-time probabilistic solution of boundary value problems. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 11160–11171. Curran Associates, Inc., 2021.

[34] N. Krämer and P. Hennig. Stable implementation of probabilistic ODE solvers. *Journal of Machine Learning Research*, 25(111):1–29, 2024.

[35] N. Krämer, J. Schmidt, and P. Hennig. Probabilistic numerical method of lines for time-dependent partial differential equations. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 625–639. PMLR, 2022.

[36] A. S. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021.

[37] M. Lange-Hegermann. Algorithmic linearly constrained Gaussian processes. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 2137–2148. Curran Associates, Inc., 2018.

[38] W. Lin, M. E. Khan, and M. Schmidt. Stein's lemma for the reparameterization trick with exponential family mixtures. *arXiv preprint arXiv:1910.13398*, 2019.

[39] W. Lin, M. Schmidt, and M. E. Khan. Handling the positive-definite constraint in the Bayesian learning rule. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020.

[40] D. Long, Z. Wang, A. Krishnapriyan, R. Kirby, S. Zhe, and M. Mahoney. AutoIP: A united framework to integrate physics into Gaussian processes. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 17–23 Jul 2022.

[41] A. Mchutchon and C. Rasmussen. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems 25 (NeurIPS)*. Curran Associates, Inc., 2011.

[42] P. Moreno-Muñoz, A. Artés, and M. Álvarez. Heterogeneous multi-output Gaussian process prediction. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*. Curran Associates, Inc., 2018.

[43] M. Padidar, X. Zhu, L. Huang, J. R. Gardner, and D. S. Bindel. Scaling Gaussian processes with derivative information using variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021.

[44] I. Pan, L. R. Mason, and O. K. Matar. Data-centric engineering: integrating simulation, machine learning and statistics. challenges and opportunities. *Chemical Engineering Science*, 249: 117271, 2022.

[45] T. Papamarkou, M. Skoularidou, K. Palla, L. Aitchison, J. Arbel, D. Dunson, M. Filippone, V. Fortuin, P. Hennig, J. M. Hernández-Lobato, A. Hubin, A. Immer, T. Karaletsos, M. E. Khan, A. Kristiadi, Y. Li, S. Mandt, C. Nemeth, M. A. Osborne, T. G. J. Rudner, D. Rügamer, Y. W. Teh, M. Welling, A. G. Wilson, and R. Zhang. Position: Bayesian deep learning is needed in the age of large-scale AI. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39556–39586. PMLR, 2024.

[46] M. Pförtner, I. Steinwart, P. Hennig, and J. Wenger. Physics-informed Gaussian process regression generalizes linear PDE solvers. *arXiv preprint arXiv:2212.12474*, 2022.

[47] L. Podina, M. T. Rad, and M. Kohandel. Conformalized physics-informed neural networks. In *ICLR 2024 Workshop on AI4Differential Equations in Science*, 2024. URL `https://openreview.net/forum?id=ZoFWS818qG`.

[48] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 2017.

[49] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[50] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2006.

[51] J. Riihimäki and A. Vehtari. Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *Proceedings of Machine Learning Research*. PMLR, 2010.

[52] H. Salimbeni and M. Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4588–4599. Curran Associates, Inc., 2017.

[53] H. Salimbeni, S. Eleftheriadis, and J. Hensman. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[54] S. Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. In *Proceedings of the 21st International Conference on Artificial Neural Networks - Volume Part II*, ICANN'11, page 151–158. Springer-Verlag, 2011.

[55] S. Särkkä and A. F. García-Fernández. Temporal parallelization of bayesian smoothers. *IEEE Transactions on Automatic Control*, 2021.

[56] S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

[57] J. Schmidt, N. Krämer, and P. Hennig. A probabilistic state space model for joint inference from differential equations and data. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 12374–12385. Curran Associates, Inc., 2021.

[58] M. Schober, D. K. Duvenaud, and P. Hennig. Probabilistic ODE solvers with Runge-Kutta means. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*. Curran Associates, Inc., 2014.

[59] E. Solak, R. Murray-smith, W. Leithead, D. Leith, and C. Rasmussen. Derivative observations in Gaussian process models of dynamic systems. In *Advances in Neural Information Processing Systems 15 (NIPS)*. MIT Press, 2002.

[60] A. Solin. *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. Doctoral thesis, Aalto University, 2016.

[61] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *Transactions on Robotics*, 34(4):1112–1127, 2018.

[62] T. Stocker. *Introduction to Climate Modelling*. Springer Science & Business Media, 2011.

[63] S. Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[64] M.-N. Tran, D. H. Nguyen, and D. Nguyen. Variational Bayes on manifolds. *Statistics and Computing*, 31:1–17, 2021.

[65] F. Tronarp, H. Kersting, S. Särkkä, and P. Hennig. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 29: 1297–1315, 2018.

[66] J. Vanhatalo, M. Hartmann, and L. Veneranta. Additive Multivariate Gaussian Processes for Joint Species Distribution Modeling with Heterogeneous Data. *Bayesian Analysis*, 15:415 – 447, 2020.

[67] C. Villacampa-Calvo, B. Zaldívar, E. C. Garrido-Merchán, and D. Hernández-Lobato. Multi-class Gaussian process classification with noisy inputs. *Journal of Machine Learning Research*, 22(1), 2021.

[68] N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3522–3526, 2013.

[69] K. Wang, O. Hamelijnck, T. Damoulas, and M. Steel. Non-separable non-stationary random fields. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 9887–9897. PMLR, 2020.

[70] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

[71] S. Wang, S. Sankaran, and P. Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024.

[72] W. J. Wilkinson, S. Särkkä, and A. Solin. Bayes-Newton methods for approximate Bayesian inference with PSD guarantees. *Journal of Machine Learning Research*, 24(83):1–50, 2023.

[73] 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*, page 1139–1146, 2012.

[74] L. Wu, G. Pleiss, and J. P. Cunningham. Variational nearest neighbor Gaussian process. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 24114–24130. PMLR, 2022.

[75] D. Zhang, L. Lu, L. Guo, and G. E. Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, 2019.

[76] M. Álvarez, D. Luengo, and N. D. Lawrence. Latent force models. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5 of *Proceedings of Machine Learning Research*, pages 9–16. PMLR, 2009.

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

# Appendices

# A Variational Approximation Derivation

## A.1 Overview of Notation

Table 4: Table of Notation

| Symbol | Size | Description |
|---|---|---|
| $N$ | – | Number of observations. |
| $Q$ | – | Number of latent functions. |
| $P$ | – | Number of latent outputs. |
| $F$ | – | Number of input features. |
| $N_\mathrm{s}$ | – | Number of spatial points. |
| $N_\mathrm{t}$ | – | Number of temporal points. |
| $d_s$ | – | Number of spatial derivatives. |
| $d_t$ | – | Number of temporal derivatives. |
| $D = d_s \cdot d_t$ | – | Total number of spatio-temporal derivatives. |
| $d$ | – | State dimension. |
| $B_\mathrm{s}$ | – | Spatial batch size. |
| $M_s$ | – | Number of spatial inducing points. |
| $\mathbf{X}$ | $N \times F$ | Input data matrix. |
| $\mathbf{X}_\mathrm{s}$ | $N_\mathrm{s} \times F$ | Spatial Locations of training data. |
| $\mathbf{X}_\mathrm{t}$ | $N_\mathrm{t}$ | Temporal locations of training data. |
| $\mathbf{x}, \mathbf{X}_n, \mathbf{X}_{t,s}$ | $F$ | Single training input. |
| $\mathbf{x}_t$ | – | Temporal axis of a single training input location $\mathbf{x}$. |
| $\mathbf{x}_s$ | $F-1$ | Spatial axes of a single training input location $\mathbf{x}$.. |
| $\mathbf{Y}$ | $N \times P$ | Output data matrix. |
| $\mathbf{Y}_n, \mathbf{Y}_{t,s}$ | $P$ | Single training output. |
| $\bar{\mathbf{f}}$ | $N_\mathrm{s} \times d$ | Filtering state. |
| $\mathbf{W}$ | $(P \times D) \times (Q \times D)$ | Mixing matrix between $Q$ latent GPs |
| $\bar{f}_q(\mathbf{X}_n)$ | $D$ | Random vector of the $D$ derivatives at location $\mathbf{X}_n$ |
| $\mathbf{F}_n$ | $(P \times D)$ | Output of linearly mixed GPs. |
| $g : \mathbb{R}^{P \cdot D} \to \mathbb{R}$ | – | Differential equation defined using $D$ spatio-temporal derivatives and $P$ outputs/states. |
| $\mathbf{Z}_\mathrm{s}$ | $\mathbb{R}^{M_s \times (F-1)}$ | Spatial Inducing Points. |
| $\mathbf{K}_s(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s})$ | $N_\mathrm{s} \times N_\mathrm{s}$ | Spatial Kernel. |
| $\mathbf{K}_t(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t})$ | $N_\mathrm{t} \times N_\mathrm{t}$ | Temporal Kernel. |
| $\mathbf{K}(\mathbf{X}, \mathbf{X})$ | $\mathrm{N} \times \mathrm{N}$ | Spatio-Temporal Kernel. |
| $\bar{\mathbf{K}} = \mathcal{D}\,\mathbf{K}(\mathbf{X}, \mathbf{X})\,\mathcal{D}^*$ | $(\mathrm{N} \cdot D) \times (\mathrm{N} \cdot D)$ | Spatio-temporal kernel over all N locations and $D$ derivatives. |
| $\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_t, \mathbf{X}_t)$ | $(N_\mathrm{t} \times d_t) \times (N_\mathrm{t} \times d_t)$ | Gram matrix over temporal derivatives. |
| $\mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_t, \mathbf{X}_t)$ | $(N_\mathrm{s} \times d_s) \times (N_\mathrm{s} \times d_s)$ | Gram matrix over spatial derivatives. |

## A.2 Layout of Vectors and Matrices

We use a numerator layout for derivatives. Let $Q$ denote the number of independent latent functions and $D$ the number of derivatives computed, and let $f_{q,d}$ denote the latent GP for $d$'th derivative of the $q$'th latent function. We will need to keep track of the permutation of our data *w.r.t.* to space, time, and latent functions. Inspired by 'row-major' and 'column-major' layouts, we will use the following terminology that describes the ordering of the data across latent functions and time and space:

- **latent-data:** $\mathbf{F} = \mathbf{F}_{\text{ld}} = [\mathbf{F}_1(\mathbf{X}), \cdots, \mathbf{F}_Q(\mathbf{X})]$ with $\mathbf{F}_q(\mathbf{X}) = [\mathbf{f}_{q,1}(\mathbf{X}), \cdots, \mathbf{f}_{q,D}(\mathbf{X})]$ which is ordered by stacking each of the latent functions on top of each other.
- **data-latent:** $\mathbf{F}_{\text{dl}} = [\mathbf{F}_1(\mathbf{X}_n), \cdots, \mathbf{F}_Q(\mathbf{X}_n)]_n^N$ which is ordered by stacking the latent functions evaluated at each data point across all data points.
- **time-space:** $[\mathbf{f}(\mathbf{X}_\mathrm{t}^{(\mathrm{st})})]_\mathrm{t}^{N_\mathrm{t}}$ which is ordered by stacking each of the input points at each time point on top of each other. This is only applicable when there is a single latent function.
- **latent-time-space:** $[\mathbf{F}_1(\mathbf{X}_1^{(\mathrm{st})}), \cdots, \mathbf{F}_1(\mathbf{X}_{N_\mathrm{t}}^{(\mathrm{st})}), \cdots, \mathbf{F}_Q(\mathbf{X}_1^{(\mathrm{st})}), \cdots, \mathbf{F}_Q(\mathbf{X}_{N_\mathrm{t}}^{(\mathrm{st})})]$

- **time-latent-space:** $[\mathbf{F}_1(\mathbf{X}_\mathrm{t}^{(\mathrm{st})}), \cdots, \mathbf{F}_Q(\mathbf{X}_\mathrm{t}^{(\mathrm{st})})]_\mathrm{t}^{N_\mathrm{t}}$

The default order will be latent-data (and latent-time-space for spatio-temporal problems). Since all of these are just simple permutations of each other, there exists a permutation matrix that permutes between any two of the layouts above. We use the function $\pi$ to denote a function that performs this permutation such that:

$$\begin{aligned}\mathbf{F}_{\text{dl}} &= \pi_{\text{ld}\to\text{dl}}(\mathbf{F}_{\text{ld}})\\ \mathbf{F}_{\text{ld}} &= \pi_{\text{dl}\to\text{ld}}(\mathbf{F}_{\text{dl}})\end{aligned} \tag{24}$$

### A.3 Timeseries Setting - Single Latent Function

Let $\mathbf{X} \in \mathbb{R}^{N\times D}, \mathbf{Y} \in \mathbb{R}^{N\times P}$ be input-output observations across $P$ outputs, where $N = N_\mathrm{t}$. For now, we only consider the case where $Q = 1$. We assume that $f$ has a state-space representation, and we denote its state with its $D$ time derivatives as $\mathbf{F}(\mathbf{X}) = [\mathbf{f}(\mathbf{X}), \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}}, \frac{\partial^2 \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}^2}, \cdots]$ in latent-data format. The vector $\mathbf{F}(\mathbf{X})$ is of dimension $(N \times D)$. We also use the notation $\mathbf{F}_n = \mathbf{F}(\mathbf{X}_n)$, which is a $D$-dimensional vector of the derivatives at location $\mathbf{X}_n$. The joint model is

$$\left[\prod_n^N p(\mathbf{Y}_n \,|\, \mathbf{F}_n, \mathrm{DE})\right] p(\mathbf{F}). \tag{25}$$

At this point, we place no particular restriction on the form of the likelihood, aside from decomposing across $N$. The prior $p(\mathbf{F})$ is a multivariate GP of dimension $N \times D$

$$p(\mathbf{F}) = \mathrm{N}\,(\,\mathbf{F} \mid \mathbf{0},\, \mathcal{D}\,\mathbf{K}(\mathbf{x},\mathbf{x})\,\mathcal{D}^*\,) \tag{26}$$

Let $q(\mathbf{F})$ be a free-form multivariate Gaussian of the same dimension as $p(\mathbf{F})$ then the corresponding ELBO is:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q(\mathbf{F})}\left[\log \frac{p(\mathbf{Y}\,|\,\mathbf{F},\mathrm{DE})\,p(\mathbf{F})}{q(\mathbf{F})}\right],\\ &= \mathbb{E}_{q(\mathbf{F})}\left[\log p(\mathbf{Y}\,|\,\mathbf{F},\mathrm{DE})\right] - \mathcal{KL}\left[q(\mathbf{F})\,||\,p(\mathbf{F})\right],\\ &= \underbrace{\sum_n^N \mathbb{E}_{q(\mathbf{F}_n)}\left[\log p(\mathbf{Y}_n\,|\,\mathbf{F}_n,\mathrm{DE})\right]}_{\text{ELL}} - \underbrace{\mathcal{KL}\left[q(\mathbf{F})\,||\,p(\mathbf{F})\right]}_{\text{KL}},\end{aligned} \tag{27}$$

and the marginal $q(\mathbf{F}_n)$ is a $D$-dimensional Gaussian corresponding to the $n$'th observation. The natural gradients are

$$\left.\widetilde{\boldsymbol{\lambda}} \leftarrow (1-\beta)\,\widetilde{\boldsymbol{\lambda}} + \beta\,\frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}}\right\} \text{ Surrogate likelihood update} \tag{28}$$

$$\left.\boldsymbol{\lambda} \leftarrow \widetilde{\boldsymbol{\lambda}} + \boldsymbol{\eta}\right\} \text{ Surrogate model update} \tag{29}$$

where $\widetilde{\boldsymbol{\lambda}} = \left[[\widetilde{\boldsymbol{\lambda}}]_1, [\widetilde{\boldsymbol{\lambda}}]_2\right]^\top$ and $[\widetilde{\boldsymbol{\lambda}}]_1$ is an $(N \times D)$ vector and $[\widetilde{\boldsymbol{\lambda}}]_2$ an $(N \times D) \times (N \times D)$ matrix. Eqn. (29) is a sum of natural parameters, and so is the conjugate Bayesian update. Naively computing this would yield no computation speed up as the computation cost would be cubic $\mathcal{O}(N^3)$. However, the natural parameters of the likelihood ($\widetilde{\boldsymbol{\lambda}}$) are guaranteed to be block diagonal, one block per data point (if $\widetilde{\boldsymbol{\lambda}}_0$ is initialised as so). This immediately implies that Eqn. (29) can be computed using efficient Kalman filter and smoothing algorithms. The structure of $\widetilde{\boldsymbol{\lambda}}$ depends on the gradient of the expected log-likelihood $\frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}}$. Expanding this out

$$\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}]_2} = \sum_n^N \frac{\partial \mathbb{E}_{q(\mathbf{F}_n)}\left[\log p(\mathbf{Y}_n\,|\,\mathbf{F}_n,\mathrm{DE})\right]}{\partial [\boldsymbol{\mu}]_2} = \sum_n^N \widetilde{\boldsymbol{\mu}}_n \tag{30}$$

where each component $\widetilde{\boldsymbol{\mu}}_n$ is a $(N \times D) \times (N \times D)$ matrix that only has $D \times D$ non-zero entries; as these are the only elements that directly affect $\mathbf{F}_n$. Collecting all these submatrices into a block diagonal matrix, we have a matrix in data-latent format, however, $\frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}}$ is in latent-data, and so all we need to do is permute by $\mathbf{P}$:

$$\frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}} = \pi_{\text{dl}\to\text{ld}}\left(\,\mathrm{blkdiag}[\,\widetilde{\boldsymbol{\mu}}_1, \cdots, \widetilde{\boldsymbol{\mu}}_N\,]\,\right). \tag{31}$$

Converting from natural to moment paramerisation the surrogate update is:

$$\begin{aligned} q(\mathbf{F}) &\propto \mathrm{N}\left(\widetilde{\mathbf{Y}} \mid \mathbf{F}, \widetilde{\mathbf{V}}\right) p(\mathbf{F}) \\ &= \left[\prod_n^N \mathrm{N}\left(\widetilde{\mathbf{Y}}_n \mid \mathbf{F}_n, \widetilde{\mathbf{V}}_n\right)\right] p(\mathbf{F}) \end{aligned} \tag{32}$$

where $\widetilde{\mathbf{Y}}_n$ is a $D$-dimensional vector, and $\widetilde{\mathbf{V}}_n$ is a $D \times D$ matrix, and efficient Kalman filtering and smoothing algorithms can be used to compute the surrogate model update. Substituting $q(\mathbf{F})$ back into the ELBO it further simplifies:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{F})}\left[\log \frac{p(\mathbf{Y} \mid \mathbf{F}, \mathrm{DE}) \cancel{p(\mathbf{F})} p(\widetilde{\mathbf{Y}} \mid \widetilde{\mathbf{V}})}{\mathrm{N}\left(\widetilde{\mathbf{Y}} \mid \mathbf{F}, \widetilde{\mathbf{V}}\right) \cancel{p(\mathbf{F})}}\right] \\ &= \sum_n^N \mathbb{E}_{q(\mathbf{F}_n)}\left[\log p(\mathbf{Y}_n \mid \mathbf{F}_n, \mathrm{DE})\right] - \sum_n^N \mathbb{E}_{q(\mathbf{F}_n)}\left[\log \mathrm{N}\left(\widetilde{\mathbf{Y}}_n \mid \mathbf{F}_n, \widetilde{\mathbf{V}}_n\right)\right] + \log p(\widetilde{\mathbf{Y}} \mid \widetilde{\mathbf{V}}) \end{aligned} \tag{33}$$

each term can be computed efficiently as the by-product of the Kalman filtering and smoothing algorithm used to compute $q(\mathbf{F})$.

### A.4 Timeseries Setting - Multiple Latent Functions

We now generalise the previous section to handle multiple independent latent functions, *i.e.* $\mathrm{Q} > 0$. The model prior now has the form

$$p(\mathbf{F}) = \prod_q^{\mathrm{Q}} p(\mathbf{F}_q) \tag{34}$$

where $p(\mathbf{F}_q)$ is a prior over $\mathbf{f}_{q,1}$ and its $D$ partial deriatives. We consider two approaches: a mean-field approximate posterior and a full Gaussian.

The first approach defined mean-field approximate posterior $q(\mathbf{F}) \triangleq \prod_q^{\mathrm{Q}} q(\mathbf{F}_q)$ where each $q(\mathbf{F}_q)$ is a free-form Gaussian of dimension $(N \times D)$. The natural gradient updates are now simply applied to each component $q(\mathbf{F}_q)$ separately, and we essentially follow the update set out in App. A.3.

The second approach is a full-Gaussian approximate posterior where $q(\mathbf{F})$ is a $(\mathrm{Q} \times D \times N)$-dimensional free-form Gaussian. In this case the ELL is

$$\mathrm{ELL} = \sum_n^N \mathbb{E}_{q(\mathbf{F}_n)}\left[\log p(\mathbf{Y}_n \mid \mathbf{F}_n, \mathrm{DE})\right] \tag{35}$$

where $q(\mathbf{F}_n)$ is of dimension $(\mathrm{Q} \times D)$. This implies that the gradient of the ELL $\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}]_2} = \sum_n^N \widetilde{\boldsymbol{\mu}}_n$ where $\widetilde{\boldsymbol{\mu}}_n$ now has $(\mathrm{Q} \times D) \times (\mathrm{Q} \times D)$ non-zero entries. Switching to moment parameterisation

$$q(\mathbf{F}) \propto \left[\prod_n^N \mathrm{N}\left(\widetilde{\mathbf{Y}}_n \mid \mathbf{F}_n, \widetilde{\mathbf{V}}_n\right)\right] p(\mathbf{F}) \tag{36}$$

where $\widetilde{\mathbf{Y}}_n$ is of dimension $(\mathrm{Q} \times D)$ and $\widetilde{\mathbf{V}}_n$ is $(\mathrm{Q} \times D) \times (\mathrm{Q} \times D)$. We can still use state-space algorithms by simply stacking the states corresponding to each $\mathbf{F}_q$ [56].

### A.5 Spatio-temporal Data - Single Latent Function

We now turn to the spatio-temporal setting where $\mathbf{X}, \mathbf{Y}$ are spatio-temporal observations on a spatio-temporal grid ordered in time-space format. We now derive the conjugate variational algorithm for PHYSS-SVGP and PHYSS-SVGP$_{\mathrm{H}}$. The algorithms for PHYSS-GP and PHYSS-VGP$_{\mathrm{H}}$ are recovered as special cases when $\mathbf{Z} = \mathbf{X}_{\mathrm{S}}$.

### A.5.1 Spatial Derivative Inducing Points

We follow the standard sparse variational GP procedure and augment that prior with inducing points $\mathbf{U} = \mathcal{D}\,\mathbf{u}$ at locations $\mathbf{Z}$. We require that the inducing points are defined on a spatial-temporal grid at the same temporal points as the data $\mathbf{X}$, such that $\mathbf{Z} = [\mathbf{Z}_t]_t^{N_\mathrm{t}}$. This is required to ensure Kronecker structure between the inducing points and the data. The joint model is

$$p(\mathbf{Y} \mid \mathbf{F})\, p(\mathbf{F} \mid \mathbf{U})\, p(\mathbf{U}) \tag{37}$$

where

$$\begin{aligned} p(\mathbf{U}) &= \mathrm{N}\left(\mathbf{U} \mid \mathbf{0},\, \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\right), \\ p(\mathbf{F} \mid \mathbf{U}) &= \mathrm{N}\left(\mathbf{F} \mid \mu_{\mathbf{F}\mid\mathbf{U}},\, \Sigma_{\mathbf{F}\mid\mathbf{U}}\right) \end{aligned} \tag{38}$$

and the conditional mean and covariance are given by

$$\begin{aligned} \mu_{\mathbf{F}\mid\mathbf{U}} &= \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\right] \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\right]^{-1} \mathbf{U} \\ &= \left[\mathbf{I} \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\, (\mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s}))^{-1}\right] \mathbf{U}, \end{aligned} \tag{39}$$

and

$$\begin{aligned} \Sigma_{\mathbf{F}\mid\mathbf{U}} =& \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s})\right] \\ &- \left[\mathbf{K}_{\mathbf{X},\mathbf{Z}}^{\otimes}\right] \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\right]^{-1} \left[\mathbf{K}_{\mathbf{X},\mathbf{Z}}^{\otimes}\right]^\top \end{aligned} \tag{40}$$

where $\mathbf{K}_{\mathbf{X},\mathbf{Z}}^{\otimes} = \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\right]$ which simplifies to

$$\Sigma_{\mathbf{F}\mid\mathbf{U}} = \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \left[\mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s}) - \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\, \mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s})^{-1}\, \mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{X}_\mathrm{s})\right] \tag{41}$$

Due to the Kronecker structure, the marginal at time $t$ only depends on the inducing points in that time slice so we can still get a CVI-style update that can be computed using a state-space model. To see why we again look at the Jacobian of the ELL: $\frac{\partial \mathrm{ELL}}{\partial [\boldsymbol{\mu}]_2} = \sum_n^N \widetilde{\boldsymbol{\mu}}_n$ where $\widetilde{\boldsymbol{\mu}}_n$ now has $(D \times M) \times (D \times M)$ non-zero entries, which corresponding to needed all $M$ spatial inducing points with there derivatives to predict at a single time point. This is similar to the time series setting, except we have now predicted in space to compute marginals of $q(\mathbf{F})$. To be complete, we write that the marginal $q(\mathbf{U})$ is

$$q(\mathbf{U}) \propto \left[\prod_t^{N_\mathrm{t}} \mathrm{N}\left(\widetilde{\mathbf{Y}}_t \mid \mathbf{F}_t,\, \widetilde{\mathbf{V}}_t\right)\right] p(\mathbf{F}) \tag{42}$$

where $\widetilde{\mathbf{Y}}_t$ and $\mathbf{F}_t$ are vectors of dimension $(D \times M)$, and $\widetilde{\mathbf{V}}_t$ is a matrix of dimension $(D \times M) \times (D \times M)$. The marginals $q(\mathbf{U}_t)$, and the corresponding marginal likelihood $p(\widetilde{\mathbf{Y}} \mid \widetilde{\mathbf{V}})$ can be computed by running a Kalman filter and smoother in $\mathcal{O}(N_\mathrm{t} \cdot (M_s \cdot d_s \cdot d)^3)$. The marginal $q(\mathbf{F}) = \mathrm{N}\left(\mathbf{F} \mid \mu_{\mathbf{F}},\, \Sigma_{\mathbf{F}}\right)$ where

$$\mu_{\mathbf{F}} = \left[\mathbf{I} \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\, (\mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s}))^{-1}\right] \mathbf{m} \tag{43}$$

and

$$\Sigma_{\mathbf{F}} = \Sigma_{\mathbf{F}\mid\mathbf{U}} + \left[\mathbf{I} \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\, (\mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s}))^{-1}\right] \mathbf{S} \left[\mathbf{I} \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_\mathrm{s})\, (\mathbf{K}_s^{\mathcal{D}}(\mathbf{Z}_\mathrm{s}, \mathbf{Z}_\mathrm{s}))^{-1}\right]^\top \tag{44}$$

### A.5.2 Structured Approximate Posterior With Spatial Inducing Points

We now derive the algorithm for the case of the structured approximate posterior with spatial inducing points. The key is to define the free-form approximate posterior over the inducing points and their temporal derivatives and then use the model conditional to compute the spatial derivatives. The model is

$$p(\mathbf{Y} \mid \mathbf{F})\, p(\mathbf{F} \mid \mathcal{D}_\mathrm{t}\, \mathbf{u})\, p(\mathcal{D}_\mathrm{t}\, \mathbf{u}). \tag{45}$$

Each term is

$$\begin{aligned} p(\mathcal{D}_\mathrm{t}\, \mathbf{u}) &= \mathrm{N}\left(\mathcal{D}_\mathrm{t}\, \mathbf{u} \mid \mathbf{0},\, \mathcal{D}_\mathrm{t}\, \mathbf{K}(\mathbf{Z}, \mathbf{Z})\, \mathcal{D}_\mathrm{t}^*\right), \\ p(\mathbf{F} \mid \mathcal{D}_\mathrm{t}\, \mathbf{u}) &= \mathrm{N}\left(\mathbf{F} \mid \mu_{\mathbf{F}\mid\mathbf{U}_t},\, \Sigma_{\mathbf{F}\mid\mathbf{U}_t}\right) \end{aligned} \tag{46}$$

where

$$\begin{aligned} \mu_{\mathbf{F}\mid\mathbf{U}_t} &= \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s})\right] \left[\mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s(\mathbf{Z}_s, \mathbf{Z}_s)\right]^{-1} \mathcal{D}_\mathrm{t}\, \mathbf{u} \\ &= \left[\mathbf{I} \otimes \widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s})\, \mathbf{K}_s(\mathbf{Z}_s, \mathbf{Z}_s)^{-1}\right] \mathcal{D}_\mathrm{t}\, \mathbf{u} \end{aligned} \tag{47}$$

and

$$\begin{aligned}\Sigma_{\mathbf{F}\,|\,\mathbf{U}_t} =& \left[\, \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s}) \,\right] \\ & - \left[\, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{Z}_s}^{\otimes} \,\right] \left[\, \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \mathbf{K}_s(\mathbf{Z}_s, \mathbf{Z}_s) \,\right]^{-1} \left[\, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{Z}_s}^{\otimes} \,\right]^\top\end{aligned} \tag{48}$$

where $\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{Z}_s}^{\otimes} = \mathbf{K}_t^{\mathcal{D}}(\mathbf{X}_\mathrm{t}, \mathbf{X}_\mathrm{t}) \otimes \widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_{s\mathrm{s}})$ and

$$\widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{X}_\mathrm{s}) = \begin{bmatrix} \mathbf{K}_s(\mathbf{X}_s, \mathbf{Z}_s) \\ \mathcal{D}_\mathrm{s}\, \mathbf{K}_s(\mathbf{X}_s, \mathbf{Z}_s) \end{bmatrix}. \tag{49}$$

The approximate posterior is defined as

$$q(\mathbf{F}, \mathcal{D}_\mathrm{t}\, \mathbf{u}) = p(\mathbf{F} \,|\, \mathcal{D}_\mathrm{t}\, \mathbf{u})\, q(\mathcal{D}_\mathrm{t}\, \mathbf{u}) \tag{50}$$

where $q(\mathcal{D}_\mathrm{t}\, \mathbf{u}$ is a free-form Gaussian of dimension $(Nd \times N_\mathrm{t} \times M_s)$. The rest of the derivation simply follows App. A.5.1 by simpling substituting $\widetilde{\mathbf{K}}_s^{\mathcal{D}}(\mathbf{X}_\mathrm{s}, \mathbf{Z}_{s\mathrm{s}})$ into the corresponding conditionals. The final result is that the approximate posterior decomposes as

$$q(\mathbf{U}) \propto \left[ \prod_t^{N_\mathrm{t}} \mathrm{N}\left( \widetilde{\mathbf{Y}}_t \mid \mathbf{F}_t,\, \widetilde{\mathbf{V}}_t \right) \right] p(\mathbf{F}) \tag{51}$$

where $\widetilde{\mathbf{Y}}_t$ and $\mathbf{F}_t$ are vectors of dimension $(d_t \times M_s)$, and $\widetilde{\mathbf{V}}_t$ is a matrix of dimension $(d_t \times M_s) \times (d_t \times M_s)$. The marginals $q(\mathbf{U}_t)$, and the corresponding marginal likelihood $p(\widetilde{\mathbf{Y}} \mid \widetilde{\mathbf{V}})$ can be computed by running a Kalman filter and smoother in $\mathcal{O}(N_\mathrm{t} \cdot (M_s \cdot d)^3)$, which compared to App. A.5.1 is *not* cubic in the number of spatial derivatives.

### A.5.3 Gauss-Newton Natural Gradient Approximation

We now provide the full derivation of the Gauss-Newton approximation of the natural gradient used to ensure *p.s.d* updates. We will make use of the following identities, known as the Bonnet and Price theorems (see, [38]),

$$\frac{\partial}{\partial \mu} \mathbb{E}_{q(\mathbf{f}\,|\,\mu,\Sigma)} \left[\, \ell(\mathbf{f}) \,\right] = \mathbb{E}_{q(\mathbf{f}\,|\,\mu,\Sigma)} \left[ \frac{\partial}{\partial \mathbf{f}}\, \ell(\mathbf{f}) \right] \tag{52}$$

$$\frac{\partial}{\partial \Sigma} \mathbb{E}_{q(\mathbf{f}\,|\,\mu,\Sigma)} \left[\, \ell(\mathbf{f}) \,\right] = \frac{1}{2}\, \mathbb{E}_{q(\mathbf{f}\,|\,\mu,\Sigma)} \left[ \frac{\partial^2}{\partial \mathbf{f}\, \partial \mathbf{f}^\top}\, \ell(\mathbf{f}) \right] \tag{53}$$

which describes how to bring derivatives inside expectations. To ease notations, we work with a more general description of the model presented in the main paper, where we have multiple independent latent functions and use $T_p$ to denote likelihood-specific functions which, for example, can be used to represent DE or as the identity of standard Gaussian likelihoods. The model is

$$\begin{aligned} p(\mathbf{u}_q) &= \mathrm{N}\left( \mathbf{u}_q \mid 0,\, \mathbf{K}_q \right) \\ p(\mathbf{f}_q \,|\, \mathbf{u}_q) &= \mathrm{N}\left( \mathbf{f}_q \mid \mu_{\mathbf{f}_q \,|\, \mathbf{u}_q},\, \Sigma_{\mathbf{f}_q \,|\, \mathbf{u}_q} \right) \\ \mathbf{Y}_{n,q} &= p(\mathbf{Y}_{n,q} \,|\, T_p(\mathbf{f}_{n,1}, \ldots, \mathbf{f}_{n,Q})) \end{aligned} \tag{54}$$

where the shapes are $\mathbf{u}_q \in \mathbb{R}^M$, $\mathbf{f}_q \in \mathbb{R}^N$, $T_p : \mathbb{R}^Q \to \mathbb{R}^P$, $\mathbf{Y} \in \mathbb{R}^{N \times P}$, and $\mathbf{Y}_{n,p} \in \mathbb{R}$. The variational approximation is

$$q(\mathbf{U}) = \mathrm{N}\left( \mathbf{U} \mid \mathbf{m},\, \mathbf{S} \right) \tag{55}$$

where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_Q]$, $\mathbf{m} \in \mathbb{R}^{QM \times 1}$ and $\mathbf{S} \in \mathbb{R}^{QM \times QM}$. Let $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_Q]$. The expected log-likelihood of the variational approximation is

$$\begin{aligned} \mathrm{ELL} &= \mathbb{E}_{q(\mathbf{U})} \left[ \mathbb{E}_{p(\mathbf{F}\,|\,\mathbf{U})} \left[ \sum_{n,p} \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \\ &= \sum_{n,p} \mathbb{E}_{q(\mathbf{U})} \left[\, \mathbb{E}_{p(\mathbf{F}_n\,|\,\mathbf{U})} \left[ \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \\ &= \sum_{n,p} \mathbb{E}_{q(\mathbf{U}_k)} \left[\, \mathbb{E}_{p(\mathbf{F}_n\,|\,\mathbf{U}_k)} \left[ \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \end{aligned} \tag{56}$$

where $k = t(n)$ is the time period associated with data $\mathbf{X}_n$. Here $\mathbf{U}_k$ are the spatial inducing points at time $t(n)$ and hence $\mathbf{U}_k \in \mathbb{R}^{QM_s}$. We need to compute

$$\begin{aligned}
\frac{\partial \text{ELL}}{\partial \mathbf{S}} &= \sum_{n,p} \frac{\partial}{\partial \mathbf{S}} \mathbb{E}_{q(\mathbf{U}_k)} \left[ \mathbb{E}_{p(\mathbf{F}_n \,|\, \mathbf{U}_k)} \left[ \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \\
&= \sum_{n,p} \mathbf{P}_k \cdot \frac{\partial}{\partial \mathbf{S}_k} \mathbb{E}_{q(\mathbf{U}_k)} \left[ \mathbb{E}_{p(\mathbf{F}_n \,|\, \mathbf{U}_k)} \left[ \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \cdot \mathbf{P}_k^\top \\
&= \sum_{n,p} \mathbf{P}_k \cdot \mathbb{E}_{q(\mathbf{U}_k)} \left[ \frac{\partial^2}{\partial \mathbf{U}_k \, \partial \mathbf{U}_k{}^\top} \mathbb{E}_{p(\mathbf{F}_n \,|\, \mathbf{U}_k)} \left[ \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}_{n,p})) \right] \right] \cdot \mathbf{P}_k^\top \\
&\overset{\text{delta}}{\approx} \sum_{n,p} \mathbf{P}_k \cdot \mathbb{E}_{q(\mathbf{U}_k)} \left[ \frac{\partial^2}{\partial \mathbf{U}_k \, \partial \mathbf{U}_k{}^\top} \log p(\mathbf{Y}_{n,p} \,|\, T_p(\mathbf{F}^*_{n,p})) \right] \cdot \mathbf{P}_k^\top \\
&\overset{\text{Gauss-Newton}}{\approx} \sum_{n,p} \mathbf{P}_k \cdot \mathbb{E}_{q(\mathbf{U}_k)} \left[ \left[ \frac{\partial T_p(\mathbf{F}^*_{n,p})}{\partial \mathbf{U}_k} \right]^\top \frac{\partial^2 \log p(\mathbf{Y}_{n,p} \mid T_p)}{\partial T_p \, \partial T_p{}^\top} \left[ \frac{\partial T_p(\mathbf{F}^*_{n,p})}{\partial \mathbf{U}_k} \right] \right] \cdot \mathbf{P}_k^\top \\
&= \sum_{n,p} \mathbf{P}_k \cdot \mathbb{E}_{q(\mathbf{U}_k)} \left[ \mathbf{J}_k^\top \, \mathbf{H}_k \mathbf{J} \right] \cdot \mathbf{P}_k^\top
\end{aligned} \tag{57}$$

where the shapes are

$$\mathbf{J} \in QM_s \times 1 \tag{58}$$

$$\mathbf{H}_k \in 1 \times 1 \tag{59}$$

and $\mathbf{P}_k$ is a permutation matrix that permutes from data-latent to latent-data format. In implementation, we do not need to perform this permutation as we only require the blocks $\frac{\partial \text{ELL}}{\partial \mathbf{S}_k}$ but write it here for completeness.

### A.5.4 Optimality of Natural Gradients In Linear Models

**Theorem A.1.** *Consider a linear multi-task model of the form*

$$\begin{aligned}
f_q(\cdot) &\sim \mathcal{GP}(0, \mathbf{K}_q) \\
\widetilde{\mathbf{f}}(\mathbf{x}) &= [\mathbf{f}_q(\mathbf{x})]_{q=1}^{\mathrm{Q}} \\
\mathbf{Y}_n &= \mathbf{W} \widetilde{\mathbf{f}}(\mathbf{X}_n) + \psi \;\; \textit{where} \;\; \psi \sim \mathrm{N}\left( \mathbf{0}, \, \textit{blkdiag}\left( [\sigma_p^2]_{p=1}^{\mathrm{P}} \right) \right)
\end{aligned} \tag{60}$$

*then under a full Gaussian variational approximate posterior*

$$q(\widetilde{\mathbf{f}}) \triangleq \mathrm{N}\left( \widetilde{\mathbf{f}} \mid \mathbf{m}, \, \mathbf{S} \right) \tag{61}$$

*where* $\mathbf{m} \in \mathbb{R}^{(N \times Q) \times 1}, \mathbf{S} \in \mathbb{R}^{(N \times Q) \times (N \times Q)}$ *then the natural gradient update with a learning rate of* $1$ *recovers the optimal solution* $p(\widetilde{\mathbf{f}} \,|\, \mathbf{Y})$.

*Proof.* To prove this we first derive the natural parameters of the posterior $p(\widetilde{\mathbf{f}} \,|\, \mathbf{Y})$. We then derive the closed form expression of the natural parameter update and show that they recover that of the posterior. Let

$$\mathbf{F}(\mathbf{X}) = \widetilde{\mathbf{W}} \, \widetilde{\mathbf{f}}(\mathbf{X}) \sim \mathrm{N}\left( \mathbf{F}(\mathbf{X}) \mid \mathbf{0}, \, \widetilde{\mathbf{W}} \, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} \, \widetilde{\mathbf{W}}^\top \right) \tag{62}$$

where $\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} = \text{blkdiag}\left( [\mathbf{K}_q]_{q=1}^{\mathrm{Q}} \right)$ and $\widetilde{\mathbf{W}} = \mathbf{W} \otimes \boldsymbol{I}$ then the posterior $p(\mathbf{F}(\mathbf{X}) \,|\, \mathbf{Y})$ is Gaussian

$$p(\mathbf{F}(\mathbf{X}) \,|\, \mathbf{Y}) = \mathrm{N}\left( \mathbf{F}(\mathbf{X}) \mid \mu_{\mathbf{F} \,|\, \mathbf{Y}}, \, \Sigma_{\mathbf{F} \,|\, \mathbf{Y}} \right) \tag{63}$$

with

$$\begin{aligned}
\mu_{\mathbf{F} \,|\, \mathbf{Y}} &= \left[ \widetilde{\mathbf{W}} \, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} \, \widetilde{\mathbf{W}}^\top \right] \left[ \widetilde{\mathbf{W}} \, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} \, \widetilde{\mathbf{W}}^\top + \Phi \right]^{-1} \mathbf{Y} \\
\Sigma_{\mathbf{F} \,|\, \mathbf{Y}} &= \left[ \left[ \widetilde{\mathbf{W}} \, \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} \, \widetilde{\mathbf{W}}^\top \right]^{-1} + \Phi^{-1} \right]^{-1}.
\end{aligned} \tag{64}$$

The covariance matrix can be simplified by invoking Woodbury's identity twice

$$\begin{aligned}\Sigma_{\mathbf{F}\,|\,\mathbf{Y}} &= \left[\widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\right] - \left[\widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\right]\left[\Phi + \left[\widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\right]\right]^{-1}\left[\widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\right] \\ &= \widetilde{\mathbf{W}}\left[\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}} - \widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\left[\Phi + \widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\,\widetilde{\mathbf{W}}^\top\right]^{-1}\widetilde{\mathbf{W}}\,\widetilde{\mathbf{K}}_{\mathbf{X},\mathbf{X}}\right]\widetilde{\mathbf{W}}^\top \\ &= \widetilde{\mathbf{W}}\left[\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top + \widetilde{\mathbf{K}}^{-1}_{\mathbf{X},\mathbf{X}}\right]^{-1}\widetilde{\mathbf{W}}^\top\end{aligned} \tag{65}$$

and the mean can be expressed as

$$\begin{aligned}\mu_{\mathbf{F}\,|\,\mathbf{Y}} &= \Sigma_{\mathbf{F}\,|\,\mathbf{Y}}\,\Phi^{-1}\mathbf{Y} \\ &= \widetilde{\mathbf{W}}\left[\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top + \widetilde{\mathbf{K}}^{-1}_{\mathbf{X},\mathbf{X}}\right]^{-1}\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\mathbf{Y}.\end{aligned} \tag{66}$$

Now we can immediately read off the posterior $p(\widetilde{\mathbf{f}}\,|\,\mathbf{Y})$ as $p(\mathbf{F}(\mathbf{X})\,|\,\mathbf{Y}) = p(\widetilde{\mathbf{W}}\,\widetilde{\mathbf{f}}\,|\,\mathbf{Y})$ is simply a transformed version

$$\begin{aligned}p(\widetilde{\mathbf{f}}\,|\,\mathbf{Y}) &= \mathrm{N}\left(\widetilde{\mathbf{f}}\;|\;\left[\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top + \widetilde{\mathbf{K}}^{-1}_{\mathbf{X},\mathbf{X}}\right]^{-1}\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\mathbf{Y},\;\left[\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top + \widetilde{\mathbf{K}}^{-1}_{\mathbf{X},\mathbf{X}}\right]^{-1}\right) \\ &= \mathrm{N}\left(\widetilde{\mathbf{f}}\;|\;\mu_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}},\,\Sigma_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}}\right)\end{aligned} \tag{67}$$

whose natural parameters are

$$\boldsymbol{\lambda}_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}} = \left[\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\mathbf{Y}, -\frac{1}{2}\,\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top - \frac{1}{2}\,\widetilde{\mathbf{K}}^{-1}_{\mathbf{X},\mathbf{X}}\right]^\top. \tag{68}$$

We now derive the closed form expression of the natural gradient update with a learning rate of 1, and show that it recovers $\boldsymbol{\lambda}_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}}$. The expected log likelihood (ELL) is

$$\begin{aligned}\mathrm{ELL} &= \mathbb{E}_{q(\widetilde{\mathbf{f}})}\left[\log \mathrm{N}\left(\mathbf{Y}\;|\;\widetilde{\mathbf{W}}\,\widetilde{\mathbf{f}},\,\Phi\right)\right] \\ &= \log \mathrm{N}\left(\mathbf{Y}\;|\;\widetilde{\mathbf{W}}\,\widetilde{\mathbf{f}},\,\Phi\right) - \frac{1}{2}\,\mathrm{Tr}\left[\Phi^{-1}\,\widetilde{\mathbf{W}}\,\mathbf{S}\,\widetilde{\mathbf{W}}^\top\right].\end{aligned} \tag{69}$$

The required derivatives are

$$\begin{aligned}\frac{\partial \mathrm{ELL}}{\partial \mathbf{m}} &= -\frac{1}{2}\,\frac{\partial}{\partial \mathbf{m}}\left[(\mathbf{Y} - \widetilde{\mathbf{W}}\,\mathbf{m})^\top\,\Phi^{-1}\,(\mathbf{Y} - \widetilde{\mathbf{W}}\,\mathbf{m})\right] \\ &= \widetilde{\mathbf{W}}^\top\,\Phi^{-1}\,(\mathbf{Y} - \widetilde{\mathbf{W}}\,\mathbf{m})\end{aligned} \tag{70}$$

where the last follows because $\Phi$ is symmetric and

$$\begin{aligned}\frac{\partial \mathrm{ELL}}{\partial \mathbf{S}} &= -\frac{1}{2}\,\frac{\partial}{\partial \mathbf{S}}\left[\mathrm{Tr}\left[\Phi^{-1}\,\widetilde{\mathbf{W}}\,\mathbf{S}\,\widetilde{\mathbf{W}}^\top\right]\right] \\ &= -\frac{1}{2}\,\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top.\end{aligned} \tag{71}$$

The natural gradient is now given as

$$\begin{aligned}\frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}}} &= \begin{bmatrix}\frac{\partial \mathrm{ELL}}{\partial \mathbf{m}} - 2\,\frac{\partial \mathrm{ELL}}{\partial \mathbf{S}}^\top\mathbf{m} \\ \frac{\partial \mathrm{ELL}}{\partial \mathbf{S}}\end{bmatrix} = \begin{bmatrix}\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\,(\mathbf{Y} - \widetilde{\mathbf{W}}\,\mathbf{m}) - \widetilde{\mathbf{W}}^\top\,\Phi^{-1}\,\widetilde{\mathbf{W}}\,\mathbf{m} \\ -\frac{1}{2}\,\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top\end{bmatrix} \\ &= \begin{bmatrix}\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\,\mathbf{Y} \\ -\frac{1}{2}\,\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top\end{bmatrix}.\end{aligned} \tag{72}$$

The natural gradient update with a learning rate of 1 is

$$\boldsymbol{\lambda}_{q(\widetilde{\mathbf{f}})} = \frac{\partial \mathrm{ELL}}{\partial \boldsymbol{\mu}_{\widetilde{\mathbf{f}}\,|\,\mathbf{Y}}} + \boldsymbol{\lambda}_{p(\widetilde{\mathbf{f}})} \tag{73}$$

where $\boldsymbol{\lambda}_{p(\widetilde{\mathbf{f}})} = \left[\mathbf{0}, -\frac{1}{2}\mathbf{K}^{-1}\right]^\top$ are the natural parameters of the prior $p(\widetilde{\mathbf{f}})$, hence after the update the natural parameters are

$$\boldsymbol{\lambda}_{q(\widetilde{\mathbf{f}})} = \begin{bmatrix}\widetilde{\mathbf{W}}^\top\,\Phi^{-1}\,\mathbf{Y} \\ -\frac{1}{2}\,\widetilde{\mathbf{W}}\,\Phi^{-1}\,\widetilde{\mathbf{W}}^\top - \frac{1}{2}\mathbf{K}^{-1}\end{bmatrix}. \tag{74}$$

which recover those of $p(\widetilde{\mathbf{f}}\,|\,\mathbf{Y})$, and hence we recover the optimal posterior. □

## B Further Experimental Details and Results

EKS methods were run on CPUs. State-space methods running on GPU used the parallel form of the Kalman smoother (see [55, 20]).

### B.1 An extension of AUTOIP

If one drops the requirement for state-space representations then the approximations proposed in Sec. 4 directly define approximations to the variational GP defined by Eqn. (13), and hence directly extend AUTOIP. For example on the non-linear damped pendulum in Sec. 7 we run this extension of AUTOIP with whitening and 50 inducing points for $C = 1000$ and achieve an RMSE of $0.06 \pm 0.001$ and running time of $158.16 \pm 0.34$, clearly improving the running time against AUTOIP. However the benefit of our methods is that PHYSS-GP remains linear in temporal dimensions which is vital for applications that are highly structured in time [20].

### B.2 Modelling Unknown Physics

Modelling of missing physics can be handled by parameterising unknown terms with GPs. For example take a simple non-linear pendulum

$$\frac{d^2\theta}{dt^2} + \sin(\theta) = 0. \tag{75}$$

Now consider that the the $\sin(\theta)$ is unknown and we would like to learn it. If we define the our differential equation in Eqn. (4) as

$$g = \frac{d^2 f_1}{dt^2} + f_2(t) = 0 \tag{76}$$

where both $f_1(\cdot), f_2(\cdot)$ are latent GPs that we wish to learn. We now construct 300 observations for training from the solution of Eqn. (75) across the range $[0, 30]$ and 1000 for testing. We run PHYSS-GP and compare the similarity of the learnt latent GP $f_2(\cdot)$ to the true function at the test locations and achieve an RMSE of $0.068$ indicating we have recovered the latent force/unknown physics well.

### B.3 Monotonic Timeseries

This first example showcases the effectiveness of PHYSS-GP in learning monotonic functions. Monotonicity information is expressed by regularising the first derivative to be positive at a set of collocation points [51]:

$$p(\mathbf{Y} \,|\, \mathbf{f}) = \mathrm{N}\left(\mathbf{Y} \mid \mathbf{f},\, \sigma_y^2\right), \;\; p(\mathbf{0} \,|\, \frac{\partial \mathbf{f}}{\partial t}) \quad = \Phi(\frac{\partial \mathbf{f}}{\partial t} \cdot \frac{1}{v})$$

where $\Phi(\cdot)$ is a Gaussian cumulative distribution function, and $v = 1e-1$ is a tuning parameter that controls the steepness of the step function. We plot predictive distributions of (batch) GP and PHYSS-GP in Fig. 5. The GP fits data and does not learn a monotonic function. However, using 300 collocation points, PHYSS-GP is able to include the additional information and learn a monotonic function whilst running $1.5$ times faster.

### B.4 Non-linear Damped Pendulum

All models were run using an Nvidia Titan RTX GPU and an Intel Core i5 CPU. All were optimised for $1000$ epochs using Adam [32] with a learning rate of $0.01$. Both the GP and AUTOIP had an RBF kernel (following Long et al. [40]) and PHYSS-GP used a Matérn-7/2; all with a lengthscale of $1.0$. The observation noise was initialised to $0.01$ and the collocation $0.001$. Both were fixed for the first $40\%$ of training and then released. Predictive distribution of PHYSS-GP and AUTOIP are plotted in Fig. 6.

### B.5 Curl-free Magnetic Field Strength

All models were run using an Nvidia Titan RTX GPU and an Intel Core i5 CPU. All models are run for $5000$ epochs using Adam with a learning rate of $0.01$, and use a Matérn-3/2 kernel on time,

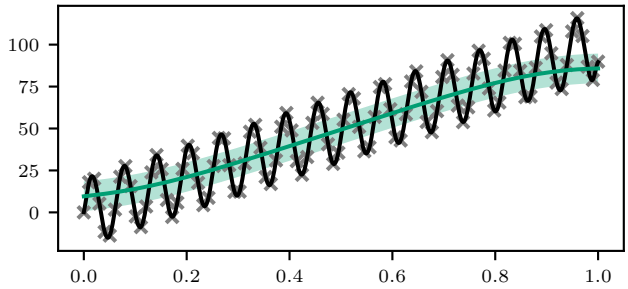


Figure 5: Predictive distributions of GP and PHYSS-GP on the monotonic function in App. B. The GP cannot incorporate monotonicity information and fits the data.

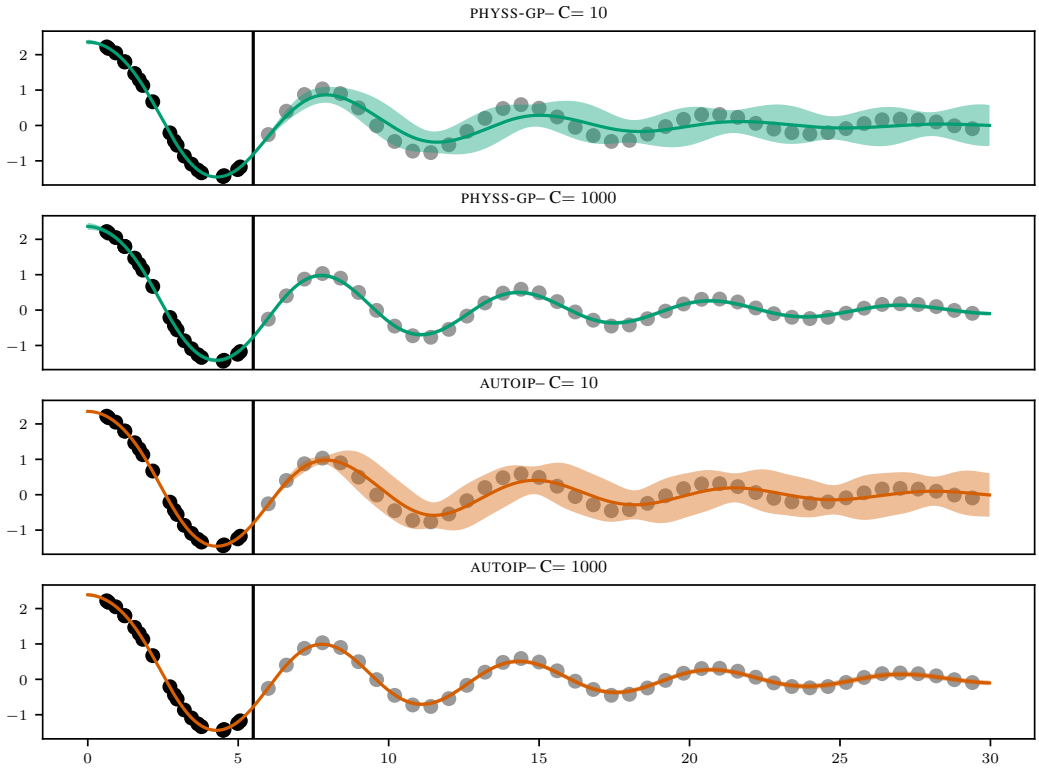


Figure 6: Predictive distributions on the Damped Pendulum.

with ARD RBF kernels on the spatial dimensions, with a lengthscale of $0.1$ across all. The Gaussian likelihood is initialised with a variance of $0.01$ and held for $40\%$ of training. All our methods used a natural gradient learning rate of $1.0$ as this is the conjugate setting.

### B.6 Diffusion-Reaction System

We use data provided by [49] under an MIT license. All models were run using an Nvidia Titan RTX GPU and an Intel Core i5 CPU. Our method PHYSS-SVGP and PHYSS-SVGP$_{\text{H}}$ use a Matérn 72 kernel on time and an RBF of space, both initialised with a lengthscale of $0.1$. We place the collocation points on a regular grid of size $20 \times 10$ and use $M_s = 20$ spatial inducing points. We pretrain for $100$ iterations using a natural gradient learning rate of $0.01$ and after use a learning rate $0.1$ for the remaining $19000$ iterations. AUTOIP uses a RBF kernel on both time and space with a lengthscale of $0.1$. We place the collocation points on a regular grid of size $10 \times 10$. All models use Adam with a learning rate $0.001$ and train for a total of $20000$ iterations.

### B.7 Ocean Currents

Our method PHYSS-SVGP$_{\text{H}}$ was run using an Nvidia Titan RTX GPU and an Intel Core i5 CPU. We ran for $10000$ iterations, using Adam with a learning rate of $0.01$. For natural gradients with used a learning rate of $0.1$. We used a Matérn-3/2 kernel on time and RBF kernels on both spatial dimensions with lengthscales $[24.0, 1.0, 1.0]$. We used $100$ spatial inducing points and a spatial mini-batch size of $10$.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have both theoretically (Theorem 3.1), and empirically demonstrated our claims (Sec. 7). We provide computational complexities for all proposed models showing a linear in the temporal dimension performance.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: A discussion of two main limitations (collocation method, and spatial scaling) and future work is provided in the conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The assumptions of Theorem 3.1 are provided in Sec. 3 with full proof (correct to the best of our knowledge) provided in the appendix App. A.5.4.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details, sufficient to reproduce the results, are provided both in the main paper Sec. 7 and the appendix App. B. Code reproducing all methods and results is provided in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code for data downloading and processing for train test splits, as well as code reproducing all methods and results is provided in the supplementary material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All required details are provided in App. B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We followed established published results for experimental setups, which did not include error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Specific details on the CPUs and GPUs used for experiments are provided in App. B.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and we conform with this in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the benefits of physics informed machine learning in our introduction. We do not envisage any negative societal implications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing methods and datasets used are open and cited throughout. Any licenses are explicitly stated in the appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release our code under CC-BY 4.0.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.