Factorial SDE for Multi-Output Gaussian Process Regression

Daniel P. Jeong
Machine Learning Department
School of Computer Science
Carnegie Mellon University
danielje@cs.cmu.edu

Seyoung Kim
Computational Biology Department
School of Computer Science
Carnegie Mellon University
sssykim@cs.cmu.edu

Abstract

Multi-output Gaussian process (GP) regression has been widely used as a flexible nonparametric Bayesian model for predicting multiple correlated outputs given inputs. However, the cubic complexity in the sample size and the output dimensions for inverting the kernel matrix has limited their use in the large-data regime. In this paper, we introduce the factorial stochastic differential equation as a representation of multi-output GP regression, which is a factored state-space representation as in factorial hidden Markov models. We propose a structured mean-field variational inference approach that achieves a time complexity linear in the number of samples, along with its sparse variational inference counterpart with complexity linear in the number of inducing points. On simulated and real-world data, we show that our approach significantly improves upon the scalability of previous methods, while achieving competitive prediction accuracy.

1 INTRODUCTION

Multi-output Gaussian process (GP) regression models have been widely used as nonparametric Bayesian models for modeling correlated multivariate outputs given inputs under uncertainty. They have been applied to many real-world problems, including inferring patient-state trajectories from longitudinal electronic health records (Ghassemi et al., 2015; Futoma et al., 2017; Cheng et al., 2020), analyzing neural activity in the brain (Marquand et al., 2014; Rutten et al., 2020), and modeling genotype$\times$environment interactions (Cuevas et al., 2017). Several types of multi-output GP regression have been proposed, such as the intrinsic models of coregionalization (IMC; Goovaerts, 1997; Bonilla et al., 2007), linear models of coregionalization (LMC; Goulard and Voltz, 1992), collaborative multi-output GPs (Nguyen and Bonilla, 2014), convolved GPs (Álvarez and Lawrence, 2011), and mixed-effects GPs (Wang and Khardon, 2012; Yoon et al., 2022). The well-known cubic complexity of exact posterior inference in both the number of samples and the number of outputs presents a major challenge in applying multi-output GP regression to large-scale data.

To reduce this computational cost in multi-output GP regression, approximate inference methods with sparse inducing points have been widely used (Titsias, 2009; Hensman et al., 2013). They reduced the time cost to cubic dependence on the number of inducing points and reduced the cubic dependence on the number of outputs to linear (van der Wilk et al., 2020; Yoon et al., 2022).

On the other hand, for single-output GP regression, recent works have shown that exact inference that scales linearly in the number of samples is possible, when a stationary GP with one-dimensional inputs is transformed into its corresponding stochastic differential equation (SDE; Särkkä and Hartikainen, 2012; Grigorievskiy et al., 2017; Särkkä and Solin, 2019). This approach has been further extended to sparse variational inference with complexity linear in the number of inducing points (Adam et al., 2020). This motivates the problem of identifying the SDE representation of multi-output GPs to further improve upon the existing sparse variational inference methods, such that exact and approximate inference scales linearly in the number of samples or inducing points.

In this paper, we present the SDE representation for a class of multi-output GPs, IMC and LMC with one-dimensional inputs, that has a factorial structure resembling that of factorial hidden Markov models (HMMs; Ghahramani and Jordan, 1997). We propose a structured mean-field variational inference strategy (Saul and Jordan, 1995; Blei et al., 2017) that exploits this factorial structure for linear-time approximate inference and derive its sparse variational inference counterpart that scales linearly in the number of

Proceedings of the 26[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

inducing points. In addition, we present efficient algorithms for handling the block-banded structure in the resulting variational parameters. On simulated and real-world data, we empirically show that our approach significantly reduces the runtime, while achieving competitive prediction accuracy, compared to the existing multi-output GP regression models with sparse variational inference.

2 BACKGROUND

Multi-Output GP Regression. Given univariate outputs $\boldsymbol{y} = [y^{(1)}, \ldots, y^{(T)}]^T$ at T distinct time points $\boldsymbol{t} = [t^{(1)}, \ldots, t^{(T)}]^T$, single-output GP regression (Rasmussen and Williams, 2005) assumes

$$y^{(i)} = f(t^{(i)}) + \epsilon^{(i)}, \quad f(t) \sim \mathcal{GP}(0, k(t, t')), \tag{1}$$

where $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$, and $k : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is a covariance function for the GP prior on the latent function $f(t)$.

When the outputs are multivariate, a wide class of multi-output GP regression models have been proposed (see Álvarez et al. (2012), van der Wilk et al. (2020) for a review). In this paper, we focus on two specific models: the LMC (Goulard and Voltz, 1992) and the IMC (Goovaerts, 1997; Bonilla et al., 2007), which is a special case of the LMC. The LMC can be viewed as a special case of the convolved GP (Álvarez and Lawrence, 2011) and a generalization of the collaborative multi-output GP (Nguyen and Bonilla, 2014) and mixed-effects GP (Wang and Khardon, 2012; Yoon et al., 2022). The LMC models the correlation among multiple outputs with a sum of multiple independent separable kernels. Suppose that we observe a sequence of P-dimensional outputs $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}] \in \mathbb{R}^{P \times T}$ at T time points $\boldsymbol{t} = [t^{(1)}, \ldots, t^{(T)}]^T$. LMC assumes that

$$\boldsymbol{y}^{(i)} = \boldsymbol{f}(t^{(i)}) + \boldsymbol{\epsilon}^{(i)}, \quad \boldsymbol{f}(t) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{K}(t, t')), \tag{2}$$

where $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_P)$, and $\boldsymbol{K} : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^{P \times P}$ is a matrix-valued covariance function

$$\boldsymbol{K}(t, t') = \sum_{\ell=1}^{L} k_{\text{in}}^{\ell}(t, t') \cdot \boldsymbol{K}_{\text{out}}^{\ell}, \tag{3}$$

where $k_{\text{in}}^{\ell}(t, t')$ for all $\ell \in [L] = \{1, \ldots, L\}$ is a covariance function defined over the inputs, and $\boldsymbol{K}_{\text{out}}^{\ell} \succeq 0$ is a $P \times P$ matrix modeling the output correlations. The IMC is a special case of the LMC when $L = 1$.

An alternative way to construct the LMC is to linearly combine L independent latent GPs (van der Wilk et al., 2020):

$$\begin{gathered} \boldsymbol{y}^{(i)} = \boldsymbol{f}(t^{(i)}) + \boldsymbol{\epsilon}^{(i)}, \quad \boldsymbol{f}(t) = \sum_{\ell=1}^{L} \boldsymbol{w}_\ell g_\ell(t), \\ g_\ell(t) \sim \mathcal{GP}(0, k_{\text{in}}^{\ell}(t, t')), \quad \forall \ell \in [L], \end{gathered} \tag{4}$$

where $\boldsymbol{w}_\ell \in \mathbb{R}^P$ induces output correlation. It follows that $\boldsymbol{K}_{\text{out}}^{\ell} = \boldsymbol{w}_\ell \boldsymbol{w}_\ell^T \succeq 0$, which is a rank-1 matrix.

Exact posterior inference in IMC and LMC is expensive for large T and P, as $\mathcal{O}(P^3T^3)$-time cost is incurred from inverting the $PT \times PT$ kernel matrix. Sparse variational inference with inducing points (Titsias, 2009) has been widely used along with stochastic optimization (Hoffman et al., 2013; Hensman et al., 2013) to reduce the time complexity for IMC and LMC to $\mathcal{O}(PLTM^2 + M^3)$ and $\mathcal{O}(PLTM^2 + LM^3)$, respectively, for M inducing points and L latent GPs (van der Wilk et al., 2020, Section 4.3.4).

SDE Representation of Single-Output GP Regression. As an alternative to sparse variational inference, the SDE representation of a single-output GP has been used for inference, because exact posterior inference is possible in linear time with Kalman filtering and smoothing (Murphy, 2012; Särkkä, 2013). To learn the model, an expectation-maximization (EM) algorithm (Dempster et al., 1977) has been used, with Bayesian smoothing in the E-step and optimization of the model parameters and kernel hyperparameters in the M-step.

For many stationary covariance functions commonly used in GP regression, a single-output GP of the form in Eq. (1) can be characterized as the exact solution (e.g., Matérn kernel) or approximate solution (e.g., squared exponential, periodic kernels) to a D-th order linear time-invariant SDE (Hartikainen and Särkkä, 2010; Särkkä et al., 2013). Collecting the derivatives of $f(t)$ into a state vector $\boldsymbol{z}(t) = [f(t), \frac{df(t)}{dt}, \ldots, \frac{d^{D-1}f(t)}{dt^{D-1}}]^T$, the SDE form of Eq. (1) is

$$\begin{aligned} d\boldsymbol{z}(t) &= \boldsymbol{A}\boldsymbol{z}(t)dt + \boldsymbol{B}d\beta(t), \\ f(t) &= \boldsymbol{U}\boldsymbol{z}(t), \end{aligned}$$

where $\beta(t)$ is a Wiener process with diffusion coefficient Q, $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ is the state-transition matrix, $\boldsymbol{B} = [0, \ldots, 0, 1]^T \in \mathbb{R}^D$ is the dispersion vector for $\beta(t)$, and $\boldsymbol{U} = [1, 0, \ldots, 0] \in \mathbb{R}^{1 \times D}$ is the output mapping that extracts the function $f(t)$ from state $\boldsymbol{z}(t)$. The exact expressions of $\boldsymbol{A}$, Q, and D depend on the choice of the kernel. For a Matérn kernel with half-integer smoothness ν, lengthscale r, and signal variance κ^2, we have

$$\begin{gathered} \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ -a_1\gamma^D & -a_2\gamma^{D-1} & \cdots & -a_{D-1}\gamma^2 & -a_D\gamma \end{bmatrix}, \\ Q = \frac{2\kappa^2\sqrt{\pi}\gamma^{2\nu}\Gamma(\nu + \frac{1}{2})}{\Gamma(\nu)}, \quad D = \lceil \nu \rceil, \end{gathered} \tag{5}$$

where $a_i = \binom{D}{i-1}$, $\gamma = \frac{\sqrt{2\nu}}{r}$, $\Gamma(\cdot)$ is the gamma function, and $\lceil \cdot \rceil$ is the ceiling function.

As the Itô process $\boldsymbol{z}(t)$ satisfies the Markov property (Øksendal, 2003), noisy observations $\boldsymbol{y}$ at time points $\boldsymbol{t}$ can

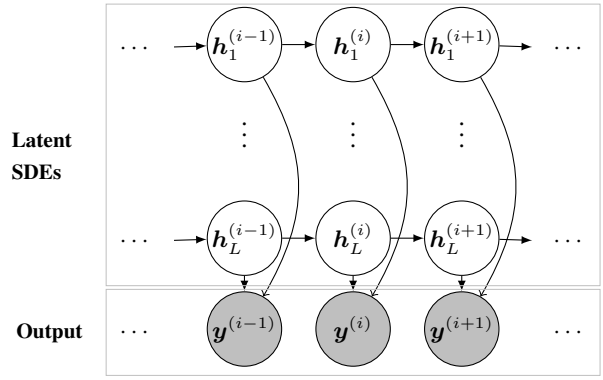


Figure 1: Graphical model of the factorial SDE for multi-output GP regression. For $\ell \in [L]$ and $i \in [T]$, $\boldsymbol{h}_\ell^{(i)}$ denotes the state vector of the ℓ-th SDE at time $t^{(i)}$, and $\boldsymbol{y}^{(i)}$ denotes the P-dimensional output at time $t^{(i)}$.

be modeled with the following discrete-time model, using the short-hand notation $\boldsymbol{z}^{(i)}$ for $\boldsymbol{z}(t^{(i)})$:

Initial state: $p\big(\boldsymbol{z}^{(1)}\big) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_\infty)$,

Transition: $p\big(\boldsymbol{z}^{(i)}|\boldsymbol{z}^{(i-1)}\big) = \mathcal{N}\big(\boldsymbol{\Psi}^{(i)}\boldsymbol{z}^{(i-1)}, \boldsymbol{\Phi}^{(i)}\big), \forall i \in [T]$

Likelihood: $p\big(y^{(i)}|\boldsymbol{z}^{(i)}\big) = \mathcal{N}\big(\boldsymbol{U}\boldsymbol{z}^{(i)}, \sigma^2\big), \forall i \in [T]$,

where $\Delta^{(i)} = t^{(i)} - t^{(i-1)}$, $\boldsymbol{\Psi}^{(i)} = e^{\Delta^{(i)}\boldsymbol{A}}$, and $\boldsymbol{\Phi}^{(i)} = \int_0^{\Delta^{(i)}} e^{(\Delta^{(i)}-\tau)\boldsymbol{A}}\boldsymbol{B}Q\boldsymbol{B}^T e^{(\Delta^{(i)}-\tau)\boldsymbol{A}^T} d\tau$. For stationary covariance functions, it is possible to compute $\boldsymbol{\Phi}^{(i)}$ without numerical integration by using the discrete Lyapunov equation, $\boldsymbol{\Phi}^{(i)} = \boldsymbol{\Sigma}_\infty - \boldsymbol{\Psi}^{(i)}\boldsymbol{\Sigma}_\infty\boldsymbol{\Psi}^{(i)^T}$ (Särkkä and Solin, 2019, Section 6.5). The closed-form expression of the steady-state covariance $\boldsymbol{\Sigma}_\infty$ can be obtained by solving the continuous Lyapunov equation $\boldsymbol{A}\boldsymbol{\Sigma}_\infty + \boldsymbol{\Sigma}_\infty\boldsymbol{A}^T + \boldsymbol{B}Q\boldsymbol{B}^T = 0$, where the exact form of $\boldsymbol{\Sigma}_\infty$ is different for different kernels (Särkkä and Solin, 2019, Sections 6.5 and 12.3).

3 FACTORIAL SDE REPRESENTATION OF MULTI-OUTPUT GP REGRESSION

In this section, we introduce a factorial SDE representation of the LMC to achieve linear-time inference in multi-output GP regression. We consider the LMC in two different forms, one in Eq. (2) and the other in Eq. (4). We show that the factorial SDE representation of the latter form leads to a significantly more compact model and more efficient inference than that of the former form.

We represent $\boldsymbol{f}(t)$ in the LMC in Eq. (4) as the following factorial SDE with the state vector $\boldsymbol{h}_\ell(t) = [g_\ell(t), \frac{dg_\ell(t)}{dt}, \ldots, \frac{d^{D-1}g_\ell(t)}{dt^{D-1}}]^T \in \mathbb{R}^D$:

$$\begin{aligned} d\boldsymbol{h}_\ell(t) &= \boldsymbol{A}_\ell\boldsymbol{h}_\ell(t)dt + \boldsymbol{B}d\beta_\ell(t), \ \forall \ell \in [L] \\ \boldsymbol{f}(t) &= \sum_{\ell=1}^{L} \boldsymbol{w}_\ell\boldsymbol{U}\boldsymbol{h}_\ell(t), \end{aligned} \tag{6}$$

where $\boldsymbol{A}_\ell \in \mathbb{R}^{D\times D}$ is the state-transition matrix corresponding to $k_{\text{in}}^\ell(t,t')$ (and identical in form to $\boldsymbol{A}$ in Eq. (5) for Matérn kernel), and $\beta_\ell(t)$ is a Wiener process with diffusion coefficient Q_{in}^ℓ corresponding to the spectral density of $k_{\text{in}}^\ell(t,t')$. Since the first component of $\boldsymbol{h}_\ell(t)$ is $g_\ell(t) = \boldsymbol{U}\boldsymbol{h}_\ell(t)$, the covariance function takes the form in Eq. (3) with $\boldsymbol{K}_{\text{out}}^\ell = \boldsymbol{w}_\ell\boldsymbol{w}_\ell^T$. For notational simplicity, throughout the paper, we assume that $k_{\text{in}}^\ell(t,t')$, $\ell \in [L]$, come from the same family of kernels with possibly different hyperparameters, such that the state vectors for all covariance functions are D-dimensional. It is straightforward to relax this assumption.

Alternatively, a different factorial SDE is obtained from the LMC representation in Eq. (2). We obtain an SDE for IMC, when the same strategy for constructing an SDE from a spatiotemporal model with GP priors with separable kernels (Glad and Ljung, 2000; Särkkä et al., 2013) is applied to IMC in Eq. (2) with $L = 1$. We extend their result to construct a factorial SDE for LMC in Eq. (2) by combining SDEs for multiple IMCs using the general-purpose algorithm for constructing an SDE from a GP prior with a sum of covariance functions (Särkkä and Solin, 2019, Section 12.3). The resulting factorial SDE is given as follows, with the state vector $\boldsymbol{z}_\ell(t) = [\boldsymbol{z}_{\ell,1}(t)^T, \ldots, \boldsymbol{z}_{\ell,P}(t)^T]^T \in \mathbb{R}^{PD}$, $\boldsymbol{z}_{\ell,p}(t) \in \mathbb{R}^D$, for all $\ell \in [L]$ and $p \in [P]$:

$$\begin{aligned} d\boldsymbol{z}_\ell(t) &= [\boldsymbol{I}_P \otimes \boldsymbol{A}_\ell]\boldsymbol{z}_\ell(t)dt \\ &\quad + [\boldsymbol{I}_P \otimes \boldsymbol{B}]d\boldsymbol{\beta}_\ell(t), \quad \forall \ell \in [L] \\ \boldsymbol{f}(t) &= \sum_{\ell=1}^{L} [\boldsymbol{I}_P \otimes \boldsymbol{U}]\boldsymbol{z}_\ell(t), \end{aligned} \tag{7}$$

where $\boldsymbol{\beta}_\ell(t)$ is a P-dimensional Wiener process with diffusion matrix $Q_{\text{in}}^\ell\boldsymbol{K}_{\text{out}}^\ell$ corresponding to $k_{\text{in}}^\ell(t,t')\boldsymbol{K}_{\text{out}}^\ell$, and $\otimes$ denotes the Kronecker product.

The main advantage of the factorial SDE in Eq. (6) over the one in Eq. (7) stems from the smaller state space with DL state variables in Eq. (6), as opposed to DPL state variables in Eq. (7). This reduction of the state-space size is achieved by modeling the output correlation via $\boldsymbol{w}_\ell$ in the linear combination of the latent states, instead of via $\boldsymbol{K}_{\text{out}}^\ell$ in the stochastic component $\boldsymbol{\beta}_\ell(t)$ of the SDE. As the smaller state space leads to more efficient inference, for the rest of this paper we focus on the factorial SDE in Eq. (6).

Our factorial SDE has a distributed state-space representation resembling that of factorial HMMs (Fig. 1; Ghahramani and Jordan, 1997). It represents the multi-output GP with a sum of separable kernels, while keeping the dynamics of all latent GPs decoupled. As we show in Section 4, this leads to linear-time inference compared to the GP representation in Eqs. (2) and (4).

Graph Kernels. Our factorial SDE can also be used to represent multi-output GPs with graph kernels (Kondor and Lafferty, 2002; Borovitskiy et al., 2021; Nikitin et al.,

2022), when the $\boldsymbol{w}_\ell$'s for $\ell \in [L]$ in Eqs. (4) and (6) correspond to a rank-L approximation to the $P \times P$ graph kernel matrix $\boldsymbol{K}_G$, i.e., $\boldsymbol{K}_G \approx \sum_{\ell=1}^{L} \boldsymbol{w}_\ell \boldsymbol{w}_\ell^T$.

Handling Multi-Dimensional Inputs. We can extend our factorial SDE to handle multi-dimensional inputs by assuming a multi-output additive regression model (Duvenaud et al., 2011; Lu et al., 2022). For C-dimensional input $\boldsymbol{x}^{(i)} = [x_1^{(i)}, \ldots, x_C^{(i)}]^T$ and output $\boldsymbol{y}^{(i)}$, we can extend Eq. (4) as $\boldsymbol{y}^{(i)} = \sum_{c=1}^{C} \boldsymbol{f}_c\big(x_c^{(i)}\big) + \boldsymbol{\epsilon}^{(i)}$, where for each $c \in [C]$, the SDE representation of $\boldsymbol{f}_c\big(x_c^{(i)}\big)$ is given by Eq. (6).

Handling Heterogeneous Outputs. To model observations with heterogeneous outputs (e.g., categorical, binary), we can assume $\boldsymbol{y}^{(i)} \sim p\big(\boldsymbol{y}|\phi\big(\boldsymbol{f}\big(t^{(i)}\big)\big)\big)$, where $\phi(\cdot)$ maps the latent function values to the appropriate parameter space via a set of inverse-link functions (Moreno-Muñoz et al., 2018), and the SDE representation of $\boldsymbol{f}\big(t^{(i)}\big)$ is given by Eq. (6). To handle the resulting non-conjugacy, we can use approximate smoothing methods such as extended Kalman smoothing (Murphy, 2012; Särkkä, 2013) and numerical integration methods such as Gauss-Hermite quadrature (Hensman et al., 2015) for approximate posterior inference.

4 VARIATIONAL INFERENCE

As in factorial HMM, exact inference and learning for the factorial SDE in Eq. (6) is possible via EM but expensive. The complete-data likelihood of finite samples for the discrete-time model of Eq. (6) factorizes as

$$p\big(\boldsymbol{y}^{(1:T)}, \boldsymbol{h}_{1:L}^{(1:T)}\big) = p\big(\boldsymbol{y}^{(1:T)}|\boldsymbol{h}_{1:L}^{(1:T)}\big) \cdot p\big(\boldsymbol{h}_{1:L}^{(1:T)}\big)$$
$$= \prod_{i=1}^{T} p\big(\boldsymbol{y}^{(i)}|\boldsymbol{h}_{1:L}^{(i)}\big) \cdot \Bigg[\prod_{\ell=1}^{L}\prod_{i=2}^{T} p\big(\boldsymbol{h}_\ell^{(i)}|\boldsymbol{h}_\ell^{(i-1)}\big) \cdot p\big(\boldsymbol{h}_\ell^{(1)}\big)\Bigg],$$

where, for simplicity, we use the short-hand notations $\boldsymbol{y}^{(1:T)} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}\}$, $\boldsymbol{h}_\ell^{(1:T)} = \{\boldsymbol{h}_\ell^{(1)}, \ldots, \boldsymbol{h}_\ell^{(T)}\}$, and $\boldsymbol{h}_{1:L}^{(1:T)} = \{\boldsymbol{h}_1^{(1:T)}, \ldots, \boldsymbol{h}_L^{(1:T)}\}$. Then, in the EM algorithm, the E-step computes the posterior $p\big(\boldsymbol{h}_{1:L}^{(1:T)}|\boldsymbol{y}^{(1:T)}\big)$ with Bayesian smoothing. As in factorial HMM, the E-step for factorial SDE is prohibitively expensive when L is large, because at each time point $t^{(i)}$, all of the latent states $\boldsymbol{h}_\ell^{(i)}$ for all $\ell \in [L]$ become dependent given the observation $\boldsymbol{y}^{(i)}$ and thus the filtering and smoothing updates have to be carried out on DL-dimensional state vectors, instead of the much smaller D-dimensional state vectors for each of the L latent SDEs.

In this section, we develop an efficient inference method for our factorial SDE representation of the multi-output GP regression. Our contribution is two-fold. First, we combine the structured mean-field algorithm previously developed for factorial HMM (Ghahramani and Jordan, 1997) and the sparse variational inference previously developed for learning an SDE model for single-output GP regression (Adam et al., 2020) into a single framework. Our approach leverages the factorized structure over L SDEs in our factorial model for efficient learning. Second, we employ the existing generic algorithms for block-tridiagonal matrices to directly exploit the block structure in the variational parameters for efficient computation, which is applicable in both single-output and multi-output settings. Overall, our approach achieves time complexity linear in T time points and M inducing points for inference, compared to cubic in T and M in the existing methods.

Structured Mean-Field Variational Inference. We integrate the structured mean-field for factorial HMM and variational inference for the SDE form of single-output GP regression as follows. As in structured mean-field for factorial HMM (Ghahramani and Jordan, 1997), we approximate the posterior $p\big(\boldsymbol{h}_{1:L}^{(1:T)}|\boldsymbol{y}^{(1:T)}\big)$ with a variational distribution that factors across the L SDEs:

$$q\big(\boldsymbol{h}_{1:L}^{(1:T)}\big) = \prod_{\ell=1}^{L} q\big(\boldsymbol{h}_\ell^{(1:T)}\big).$$

Then, for each factor $q\big(\boldsymbol{h}_\ell^{(1:T)}\big)$ above, we use the parameterization of the variational distribution used in single-output GPs (Durrande et al., 2019; Adam et al., 2020):

$$q\big(\boldsymbol{h}_\ell^{(1:T)}\big) = \mathcal{N}(\boldsymbol{m}_\ell, \boldsymbol{S}_\ell^{-1}).$$

$\boldsymbol{m}_\ell \in \mathbb{R}^{TD}$ above is the variational mean and $\boldsymbol{S}_\ell$ is the $TD \times TD$ variational precision matrix with a symmetric block-tridiagonal structure:

$$\boldsymbol{S}_\ell = \begin{bmatrix} \boldsymbol{S}_\ell^{(1)} & \boldsymbol{S}_\ell^{(2,1)^T} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{S}_\ell^{(2,1)} & \boldsymbol{S}_\ell^{(2)} & \boldsymbol{S}_\ell^{(3,2)^T} & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{S}_\ell^{(3,2)} & \boldsymbol{S}_\ell^{(3)} & \ddots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \boldsymbol{S}_\ell^{(T,T-1)^T} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{S}_\ell^{(T,T-1)} & \boldsymbol{S}_\ell^{(T)} \end{bmatrix},$$

where $\boldsymbol{S}_\ell^{(i)}$ and $\boldsymbol{S}_\ell^{(i,j)}$ for $i, j \in [T]$ are $D \times D$ matrices. This block-banded parameterization reflects the first-order Markovian structure present in the SDE for $g_\ell(t)$. We denote the collection of non-zero tridiagonal blocks as $\mathrm{BTD}(\boldsymbol{S}_\ell)$. As in Adam et al. (2020), we parameterize the distribution with the Cholesky factor $\boldsymbol{S}_\ell = \boldsymbol{R}_\ell \boldsymbol{R}_\ell^T$, where $\boldsymbol{R}_\ell$ has the same structure as the lower triangular part of $\boldsymbol{S}_\ell$ (Cao et al., 2002). We constrain the diagonal entries of $\boldsymbol{R}_\ell$ to be positive to ensure $\boldsymbol{S}_\ell \succ 0$ during optimization. The block-banded structure of $\boldsymbol{R}_\ell$ implies that we only need to keep $\mathrm{BTD}(\boldsymbol{R}_\ell)$ in memory with $\mathcal{O}(LTD^2)$ space.

Given data $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}]$ at T time points, we optimize the kernel hyperparameters and the variational param-

eters by maximizing the evidence lower bound (ELBO):

$$\begin{aligned}
&\sum_{i=1}^{T}\mathbb{E}_q\big[\log p(\boldsymbol{y}^{(i)}|\boldsymbol{h}_{1:L}^{(i)})\big] - \sum_{\ell=1}^{L}\mathrm{KL}\left(q(\boldsymbol{h}_\ell^{(1:T)})||p(\boldsymbol{h}_\ell^{(1:T)})\right)\\
&= -\frac{TP}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}||\boldsymbol{Y}-\hat{\boldsymbol{Y}}||_F^2\\
&\quad - \frac{1}{2\sigma^2}\sum_{\ell=1}^{L}||\boldsymbol{w}_\ell||^2\left(\sum_{i=1}^{T}\big[\boldsymbol{S}_\ell^{-1^{(i)}}\big]_{1,1}\right)\\
&\quad - \frac{1}{2}\sum_{\ell=1}^{L}\left[\log\frac{|\boldsymbol{S}_\ell|}{|\boldsymbol{\Lambda}_\ell|} - TD + \boldsymbol{m}_\ell^T\boldsymbol{\Lambda}_\ell\boldsymbol{m}_\ell + \mathrm{tr}(\boldsymbol{\Lambda}_\ell\boldsymbol{S}_\ell^{-1})\right],
\end{aligned}$$

where $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}^{(1)},\dots,\hat{\boldsymbol{y}}^{(T)}]$, $\hat{\boldsymbol{y}}^{(i)} = \sum_{\ell=1}^{L}\boldsymbol{w}_\ell\boldsymbol{U}\boldsymbol{m}_\ell^{(i)}$ is the prediction for the i-th sample, $\boldsymbol{\Lambda}_\ell$ is the $TD\times TD$ precision matrix of the prior $p(\boldsymbol{h}_\ell^{(1:T)})$, and $\boldsymbol{S}_\ell^{-1^{(i)}}$ is the i-th diagonal block of $\boldsymbol{S}_\ell^{-1}$.

A major bottleneck in evaluating the ELBO is in computing the terms that involve the variational precision matrix: $\{\boldsymbol{S}_\ell^{-1^{(i)}}\}_{i=1}^T$ and $\log|\boldsymbol{S}_\ell|$. To avoid $\mathcal{O}(T^3D^3)$ complexity with standard matrix operations, previous works on single-output GPs took advantage of the banded structure of $\boldsymbol{S}_\ell$, but ignored the block structure in the block-tridiagonal matrix (Durrande et al., 2019; Adam et al., 2020). Below, we show efficient methods for performing these matrix operations that directly work with the block structure in $\boldsymbol{S}_\ell$. Our approach is easy to implement, as $\boldsymbol{S}_\ell$ is stored as its blocks in memory, and is numerically more stable.

We compute the inverses $\{\boldsymbol{S}_\ell^{-1^{(i)}}\}_{i=1}^T$ in $\mathcal{O}(TD^3)$ time, using the block-by-block inversion algorithm by Reuter and Hill (2012) that directly leverages the block structures in $\boldsymbol{S}_\ell$. It recursively calculates each diagonal and off-diagonal block in $\boldsymbol{S}_\ell^{-1}$ in terms of the blocks in $\boldsymbol{S}_\ell$:

$$\begin{aligned}
\boldsymbol{S}_\ell^{-1^{(i)}} &= (\boldsymbol{S}_\ell^{(i)} - \boldsymbol{\Gamma}_i - \boldsymbol{\Omega}_i)^{-1},\\
\boldsymbol{S}_\ell^{-1^{(i,i-1)}} &= -(\boldsymbol{S}_\ell^{(i)} - \boldsymbol{\Gamma}_i)^{-1}\boldsymbol{S}_\ell^{(i,i-1)}\boldsymbol{S}_\ell^{-1^{(i-1)}},
\end{aligned} \tag{8}$$

where the $D\times D$ matrices $\boldsymbol{\Gamma}_T = \boldsymbol{0}$ and $\boldsymbol{\Gamma}_i = \boldsymbol{S}_\ell^{(i+1,i)^T}(\boldsymbol{S}_\ell^{(i+1)} - \boldsymbol{\Gamma}_{i+1})^{-1}\boldsymbol{S}_\ell^{(i+1,i)}$ for $i\in[T-1]$, and $\boldsymbol{\Omega}_1 = \boldsymbol{0}$ and $\boldsymbol{\Omega}_i = \boldsymbol{S}_\ell^{(i,i-1)}\big(\boldsymbol{S}_\ell^{(i-1)} - \boldsymbol{\Omega}_\ell^{(i-1)}\big)^{-1}\boldsymbol{S}_\ell^{(i,i-1)^T}$ for $i = 2,\dots,T$. Previous works ignored the block structure in $\boldsymbol{S}_\ell$. Instead, they treated $\boldsymbol{S}_\ell$ as a generic banded matrix of size $TD\times TD$ with bandwidth $B = 2D-1$ and computed each column with the total cost $O(TB^2)$, which lead to including some zero entries that are not part of the blocks (Durrande et al., 2019; Adam et al., 2020).

We compute $\log|\boldsymbol{S}_\ell|$ in $\mathcal{O}(TD^3)$ time, using the algorithm for block-tridiagonal matrices by Salkuyeh (2006) that again directly works with block matrices. Following Salkuyeh (2006), we express the log-determinant as

$$\log|\boldsymbol{S}_\ell| = \sum_{i=1}^{T}\log|\boldsymbol{\Pi}_i|, \tag{9}$$

Table 1: Time and space complexities of sparse variational inference for different models.

Model	Time	Space
IMC	$\mathcal{O}(PLT_bM^2 + M^3)$	$\mathcal{O}(LM + M^2)$
LMC	$\mathcal{O}(PLT_bM^2 + LM^3)$	$\mathcal{O}(LM^2)$
FSDE	$\mathcal{O}(PLTD^3)$	$\mathcal{O}(LTD^2)$
FSDE-SVI	$\mathcal{O}(PL(T_b + M)D^3)$	$\mathcal{O}(LMD^2)$

where the $D\times D$ matrices $\{\boldsymbol{\Pi}_i\}_{i=1}^T$ satisfy the recurrence relations $\boldsymbol{\Pi}_1 = \boldsymbol{0}$ and $\boldsymbol{\Pi}_i = \boldsymbol{S}_\ell^{(i)} - \boldsymbol{S}_\ell^{(i,i-1)}\boldsymbol{\Pi}_{i-1}^{-1}\boldsymbol{S}_\ell^{(i,i-1)^T}$ for $i = 2,\dots,T$. Previous work on SDE models of single-output GP regression (Durrande et al., 2019; Adam et al., 2020) computed the log-determinant by using the identity $\log|\boldsymbol{S}_\ell| = 2\log|\boldsymbol{R}_\ell| = 2\sum_k\log([\boldsymbol{R}_\ell]_k)$, where $[\boldsymbol{R}_\ell]_k$ is the k-th diagonal element of the Cholesky factor $\boldsymbol{R}_\ell$. Although their approach has the cost $\mathcal{O}(TD)$ compared to $\mathcal{O}(TD^3)$ in our approach, in practice, this difference in computation time is negligible since D is typically small. Importantly, in our experiments with the factorial SDE, we found their approach to be often numerically unstable, for larger dimensions of $\boldsymbol{R}_\ell$ due to many near-zero diagonal entries in $\boldsymbol{R}_\ell$, whereas our approach did not suffer from numerical instability because we use the recursion in Eq. (9) that honors the block structure from the Markov property.

Natural Gradient Updates. To speed up convergence, we update the variational parameters with natural gradients, again taking advantage of the block structure in $\boldsymbol{S}_\ell$, in constrast to previous approaches that considered only the banded structure in $\boldsymbol{S}_\ell$. Following Hoffman et al. (2013) and Salimbeni et al. (2018), we compute the natural gradient of the ELBO $\mathcal{L}$ with respect to the variational parameters $\boldsymbol{\chi}_\ell = [\boldsymbol{m}_\ell, \mathrm{BTD}(\boldsymbol{R}_\ell)]$ as

$$\tilde{\nabla}_{\boldsymbol{\chi}_\ell}\mathcal{L} = \left(\frac{\partial\boldsymbol{\chi}_\ell}{\partial\boldsymbol{\theta}_\ell}\right)\nabla_{\boldsymbol{\xi}_\ell}\mathcal{L}, \tag{10}$$

where $\boldsymbol{\theta}_\ell = [\boldsymbol{S}_\ell\boldsymbol{m}_\ell, -\frac{1}{2}\mathrm{BTD}(\boldsymbol{S}_\ell)]$ are the natural parameters, $\boldsymbol{\xi}_\ell = [\boldsymbol{m}_\ell, \mathrm{BTD}(\boldsymbol{m}_\ell\boldsymbol{m}_\ell^T + \boldsymbol{S}_\ell^{-1})]$ are the expectation parameters of $q(\boldsymbol{h}_\ell^{(1:T)})$, and $\nabla_{\boldsymbol{\xi}_\ell}\mathcal{L}$ is the Euclidean gradient with respect to $\boldsymbol{\xi}_\ell$.

Computing $\nabla_{\boldsymbol{\xi}_\ell}\mathcal{L} = (\frac{\partial\boldsymbol{\chi}_\ell}{\partial\boldsymbol{\xi}_\ell})^T\nabla_{\boldsymbol{\chi}_\ell}\mathcal{L}$ in Eq. (10) requires an efficient computation of $\mathrm{BTD}(\boldsymbol{R}_\ell)$ from $\mathrm{BTD}(\boldsymbol{S}_\ell^{-1})$. While previous works on single-output GP regression (Durrande et al., 2019; Adam et al., 2020) used a banded matrix algorithm to perform this computation, we use the $\mathcal{O}(TD^3)$ algorithm by Asif and Moura (2005) that directly obtains each block in $\mathrm{BTD}(\boldsymbol{R}_\ell)$ in terms of the blocks in

$\text{BTD}(\boldsymbol{S}_\ell^{-1})$ recursively:

$$\begin{aligned}
\boldsymbol{R}_\ell^{(1)} &= \text{Chol}\left((\boldsymbol{S}_\ell^{-1^{(1)}})^{-1}\right), \\
\boldsymbol{R}_\ell^{(i)} &= \text{Chol}\left(\boldsymbol{S}_\ell^{(i)} - \boldsymbol{S}_\ell^{(i+1,i)^T}(\boldsymbol{S}_\ell^{-1^{(i)}})^{-1}\boldsymbol{S}_\ell^{(i+1,i)}\right)^{-1}, \\
\boldsymbol{R}_\ell^{(i,i-1)} &= -(\boldsymbol{S}_\ell^{-1^{(i)}})^{-1}\boldsymbol{S}_\ell^{-1^{(i,i-1)}}\boldsymbol{R}_\ell^{(i-1)}. \qquad (11)
\end{aligned}$$

This algorithm is easily incorporated into Jacobian-vector and vector-Jacobian product operations within modern automatic differentiation frameworks such as TensorFlow (Abadi et al., 2016) and JAX (Bradbury et al., 2018).

Sparse Variational Inference. To further improve scalability, we modify the approach above for sparse variational inference with minibatch training (Hoffman et al., 2013; Hensman et al., 2013; Adam et al., 2020). We provide details of the sparse variational inference approach in Section A of the Supplementary Material. The computational costs of calculating the ELBO and the natural gradient updates are $\mathcal{O}(MD^3)$ for M inducing points using the algorithms in Eqs. (8), (9), and (11). With minibatches of size T_b, the complexity of inference is reduced to $\mathcal{O}(PL(T_b+M)D^3)$, which is linear in M, in contrast to the cubic dependency on M in the existing sparse variational methods for IMC and LMC. The time and space complexities of our and existing methods are summarized in Table 1.

Forecasting and Smoothing for Prediction. We make a forecasting prediction when a new time point $t^{(*)}$ is greater than the last training time point $t^{(T)}$ or the last inducing time point $s^{(M)}$. We make a smoothing prediction when $t^{(*)}$ is between two training time points $t^{(i-1)}$ and $t^{(i)}$ for some $i \in [T]$ or between two inducing time points $s^{(i-1)}$ and $s^{(i)}$ for some $i \in [M]$. For new time points $t^{(*)} \in \mathbb{R}$, the approximate posterior predictive distribution is given as

$$\begin{aligned}
p\big(\boldsymbol{y}^{(*)}|\boldsymbol{y}^{(1:T)}\big) &\approx \int p\big(\boldsymbol{y}^{(*)}|\boldsymbol{h}_{1:L}^{(*)}\big)\prod_{\ell=1}^{L} q\big(\boldsymbol{h}_\ell^{(*)}\big)d\boldsymbol{h}_{1:L}^{(*)} \\
&= \mathcal{N}\left(\sum_{\ell=1}^{L}\boldsymbol{w}_\ell \boldsymbol{U}\boldsymbol{m}_\ell^{(*)}, \sum_{\ell=1}^{L}[\boldsymbol{S}_\ell^{-1^{(*)}}]_{1,1}\boldsymbol{w}_\ell\boldsymbol{w}_\ell^T + \sigma^2\boldsymbol{I}_p\right),
\end{aligned}$$

where $\boldsymbol{m}_\ell^{(*)}$ is the mean and $\boldsymbol{S}_\ell^{-1(*)}$ is the covariance matrix of the approximate posterior distribution $q\big(\boldsymbol{h}_\ell^{(*)}\big)$. We provide the moments of $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ for forecasting and smoothing tasks in Section B of the Supplementary Material.

5 EXPERIMENTS

We compare the variational and sparse variational inference methods for our factorial SDE against those for the IMC and LMC baselines on simulated and real-world datasets.

We train all models in two different optimization settings: one in which we update both the kernel hyperparameters and the variational parameters with Adam (Kingma and Ba, 2015) and the other in which we update the kernel hyperparameters with Adam and the variational parameters with natural gradient descent. For the momentum hyperparameters in Adam, we use the default values given by Kingma and Ba (2015). For the factorial SDEs, we use the learning rate scheduler proposed by Salimbeni et al. (2018), where the learning rate for natural gradient descent is log-linearly increased from an initial learning rate to a final learning rate over a predefined number of iterations. We also use gradient clipping with a max-norm threshold of 10^4 to guard against numerical instability during training (Pascanu et al., 2013). To check for convergence, we compute the absolute percent change in successive averages of the ELBO calculated with a window size of 40 and declare convergence when we observe a total of five drops below a tolerance of 10^{-5}. We optimize all models to convergence but stop if the optimization fails to converge within 24 hours or reaches 50,000 iterations, whichever comes first. We run all experiments on AMD EPYC 7742 CPUs each with 16GB of RAM and 2.25-3.40 GHz clock speed and compute the mean absolute error (MAE) and negative log predictive density (NLPD) to assess performance. We implement the factorial SDE models in JAX (Bradbury et al., 2018) and all of the baselines in GPflow (Matthews et al., 2017).

5.1 Simulation Data

We demonstrate the accuracy and scalability of different methods on a small simulation dataset with $P = 10$ outputs and $T = 100$ samples and on a large simulation dataset with $P = 30$ outputs and $T = 10,000$ samples.

Small Simulation Dataset (SMALL-SIM). For the small dataset, we sample noisy observations from a LMC with five Matérn-$\frac{5}{2}$ latent GPs $\{g_1(t), \ldots, g_5(t)\}$, $\boldsymbol{w}_\ell \sim \mathcal{N}(0, 2\mathbf{I}_P)$ for $\ell \in [L]$, and $\sigma^2 = 0.2$ at equally spaced inputs $t \in [0, 50]$. For the latent GPs, we use signal variance $\kappa_\ell^2 = 1$ for $\ell \in [L]$ and lengthscales $[r_1, \ldots, r_5] = [0.5, 0.5, 0.75, 1, 1]$ as the kernel hyperparameters.

For extrapolation, we hold out the last 20 samples as test data, and for interpolation, we perform a five-fold cross-validation using the remaining 80 samples. Given that the dataset is small, we additionally include an *exact* inference baseline, whose kernel hyperparameters are optimized with respect to the marginal log-likelihood using L-BFGS (Liu and Nocedal, 1989), to evaluate the quality of the approximate posterior predictions. For all models, we use five latent GPs with Matérn-$\frac{5}{2}$ kernels, whose signal variances and lengthscales are initialized to 1. For the sparse models, we use an equally spaced grid of $M = 30$ inducing points and use minibatches of size $T_b = 40$. When Adam is used

Table 2: Five-fold cross-validation results on SMALL-SIM.

MODEL		INTERPOLATION		EXTRAPOLATION	
		MAE	NLPD	MAE	NLPD
Exact	L-BFGS	2.21 ± 0.28	1.16 ± 0.03	3.28 ± 0.04	1.67 ± 0.02
IMC	ADAM	2.66 ± 0.15	2.51 ± 0.05	3.70 ± 0.20	2.60 ± 0.04
IMC	NGD	2.66 ± 0.12	2.51 ± 0.05	3.63 ± 0.16	2.61 ± 0.04
LMC	ADAM	2.65 ± 0.15	2.50 ± 0.05	3.64 ± 0.17	2.58 ± 0.04
LMC	NGD	2.64 ± 0.13	2.50 ± 0.05	3.71 ± 0.23	3.71 ± 0.23
FSDE	ADAM	$\mathbf{2.30 \pm 0.33}$	$\mathbf{1.17 \pm 0.05}$	3.33 ± 0.03	1.67 ± 0.03
FSDE	NGD	2.31 ± 0.34	1.20 ± 0.04	3.30 ± 0.02	1.64 ± 0.04
FSDE-SVI	ADAM	2.85 ± 0.30	1.83 ± 0.27	3.24 ± 0.05	$\mathbf{1.77 \pm 0.06}$
FSDE-SVI	NGD	2.55 ± 0.13	2.47 ± 0.65	$\mathbf{3.23 \pm 0.04}$	2.19 ± 0.09

to optimize both the kernel hyperparameters and the variational parameters, we set both the learning rate η_1 for the hyperparameters and the learning rate η_2 for the variational parameters to 10^{-3} for all models. With natural gradients, we set η_2 to 10^{-2} for the IMC and LMC baselines, change η_2 from 10^{-4} to 10^{-2} over 4,000 iterations for the factorial SDE and from 10^{-5} to 10^{-4} over 4,000 iterations for the factorial SDE with sparse variational inference.

Figure 2 shows the posterior predictions on one of the 10 outputs after each model is trained on one of the 5 train-test splits with natural gradient updates. The predictions and uncertainty estimates given by the factorial SDEs closely match those of the exact posterior. In constrast, the IMC and LMC baselines provide significantly underconfident predictions that are overly smoothed despite having the same number of inducing points and latent GPs as the factorial SDE with sparse variational inference. These results are consistent with the findings for single-output GP regression in Adam et al. (2020) that using the SDE representation for variational inference increases the effective number of inducing points, as they become D-dimensional states instead of scalar-valued function evaluations. Table 2 summarizes the five-fold cross-validated test MAE and NLPD for all models trained in both optimization settings. It shows that the factorial SDE models consistently outperform the IMC and LMC baselines.

Large Simulation Dataset (LARGE-SIM). We generate data from a linear combination of J sinusoidal functions that are randomly shifted and scaled. With inputs t equally spaced in [0,1000], for the p-th output at t, we take a noisy observation of function $f_p(t) = \sum_{j=1}^{J} \alpha_{pj} v_j(t)$ with $\alpha_{pj} \sim \mathcal{N}(0,1)$ and $v_j(t) = \sum_{k=1}^{3} \left[2\sin(\frac{3\pi t}{500}) + c_k^{(1)} \sin\left(0.1 \cdot c_k^{(2)}(t - c_k^{(3)})\right)\right]$, where $c_k^{(1)} \sim \text{Uniform}(1,3), c_k^{(2)} \sim \text{Uniform}(-5,5)$, and $c_k^{(3)} \sim \text{Uniform}(-10,10)$. We set the observation noise to have unit variance.

For extrapolation, we hold out the last 200 samples as test data, and for interpolation, we perform a five-fold cross-validation using the remaining 9,800 samples. Since the dataset is much larger, we focus on the IMC, LMC, and the factorial SDE with sparse variational inference. For

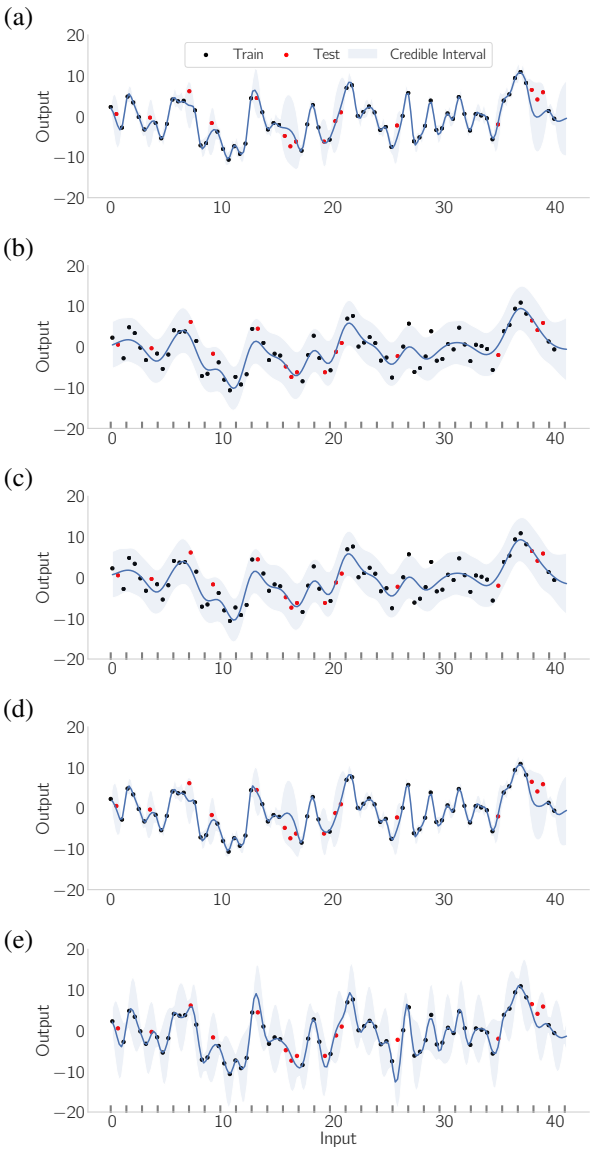


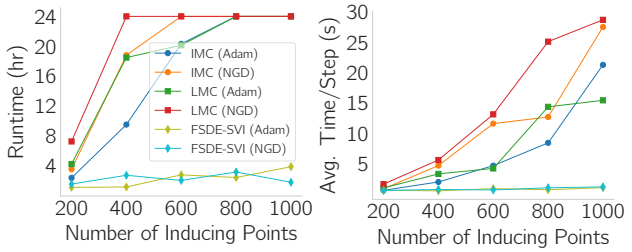
Figure 2: Comparison of the posterior predictions on one train-test split of SMALL-SIM data. (a) LMC with exact inference, (b) IMC, (c) LMC, (d) factorial SDE, and (e) factorial SDE with sparse variational inference. The gray ticks on the x-axis are the inducing time points $\boldsymbol{s}$.

all models, we use five latent GPs with Matérn-$\frac{3}{2}$ kernels, whose signal variances and lengthscales are initialized to 1. For the sparse models, we use an equally spaced grid of $M = 1,000$ inducing points and minibatches of size $T_b = 1,000$. With Adam for both the kernel hyperparameters and variational parameters, we set $\eta_1 = \eta_2 = 10^{-2}$ for all models. With natural gradients for the variational parameters, we set η_2 to 10^{-2} for the IMC and LMC baselines, and change η_2 from 10^{-4} to 10^{-2} over 500 iterations for the factorial SDE with sparse variational inference.

Table 3 shows that the factorial SDE consistently outper-

Table 3: Five-fold cross-validation results on LARGE-SIM.

MODEL		INTERPOLATION MAE	INTERPOLATION NLPD	EXTRAPOLATION MAE	EXTRAPOLATION NLPD	TIME (hr)
IMC	ADAM	**3.23 ± 0.01**	3.04 ± 0.10	6.23 ± 0.04	3.42 ± 0.09	24±0.00
IMC	NGD	3.25 ± 0.01	3.33 ± 0.13	6.48 ± 0.02	4.02 ± 0.10	24±0.00
LMC	ADAM	**3.23 ± 0.01**	3.10 ± 0.13	6.25 ± 0.04	3.47 ± 0.11	24±0.00
LMC	NGD	3.25 ± 0.01	3.35 ± 0.14	6.49 ± 0.02	4.03 ± 0.12	24±0.00
FSDE-SVI	ADAM	**3.23 ± 0.01**	**2.82 ± 0.00**	6.19 ± 0.03	**3.18 ± 0.00**	**2.62±0.71**
FSDE-SVI	NGD	**3.23 ± 0.01**	**2.82 ± 0.00**	**6.18 ± 0.08**	3.18 ± 0.01	**2.57±0.92**



(a) (b)

Figure 3: Effects of the number of inducing points on the runtime of different models on LARGE-SIM data. (a) Total runtime. (b) Average time per gradient update step.

forms the baselines on prediction accuracy in significantly less time. In particular, the factorial SDE is the only model that converges within the given time budget (see Figure S1 in the Supplementary Material for the ELBO plots over iterations).

Using one of the train-test splits, we evaluate how the runtime for each model changes as we increase the number of inducing points M from 200 to 1,000 with increments of 200 points. Figure 3 shows that, as we increase the number of inducing points, both the total runtime and the average time per gradient update grow at a significantly slower pace for the factorial SDE than for the baselines. This is because each gradient update scales cubically in M for the baselines but only linearly for the factorial SDE. Moreover, while the computational cost for using natural gradients is significantly higher than using Adam in the baseline models (Salimbeni et al., 2018, Section 3), the difference is negligibly small for the factorial SDE. Although the worst-case complexity does not change across the two optimization settings for either the baselines or the factorial SDE, in practice, the additional overhead for calculating the natural gradients does lead to a noticeable difference between the baselines and the factorial SDE. This illustrates that for a given computational budget, the linear dependence on M for factorial SDEs allows us to afford significantly more inducing points for better posterior approximation and still use the natural gradients for faster and improved convergence with little to no additional computational overhead.

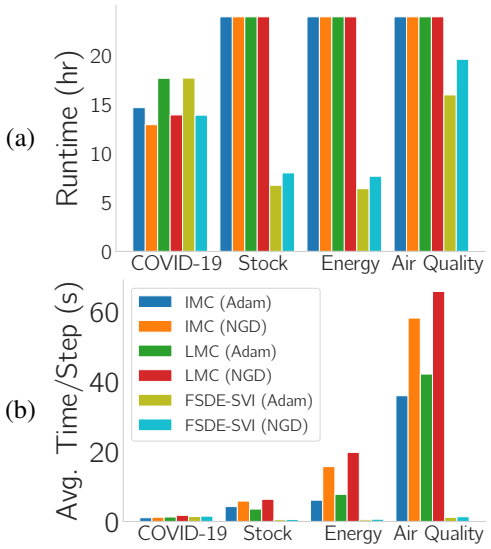


Figure 4: Comparison of the computation time for different methods on real datasets. (a) Total runtime. (b) Average time per gradient update step.

5.2 Real Data

We compare the performance of our factorial SDE with sparse variational inference against that of LMC and IMC on four real datasets: COVID-19, STOCK[1], ENERGY, and AIR QUALITY. For the latent GPs of all models, we use Matérn-$\frac{3}{2}$ kernels, whose signal variances and lengthscales are initialized to 1. We include other details of the experimental settings in Section C.2 of the Supplementary Material.

COVID-19. We use the data provided by the Center for Systems Science and Engineering at Johns Hopkins University of daily confirmed COVID-19 cases in the U.S. (Dong et al., 2020). We use the case counts for $P = 3,091$ counties in the U.S. over $T = 273$ days from July 2020 to March 2021, treating each day as an input and each county as an output. We log-normalize the case counts to be real values. We hold out the last 31 days for extrapolation and randomly take a 80-20 split to obtain the training and interpolation data. For all models, we use minibatches of size $T_b = 150$ and $M = 50$ inducing points.

STOCK. We use the closing stock prices of $P = 31$ companies in the Dow Jones Industrial Average index over $T = 3,018$ weekdays between 2006 and 2017. We standardize the stock prices to have zero-mean and unit variance. We treat each weekday as an input and the stock price of each company as an output. We hold out the last 200 days for extrapolation and randomly take a 80-20 split to obtain the training and interpolation data. For all mod-

[1] https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231

Table 4: Test results on real datasets.

MODEL		COVID-19				STOCK				ENERGY				AIR QUALITY			
		Interpolation		Extrapolation		Interpolation		Extrapolation		Interpolation		Extrapolation		Interpolation		Extrapolation	
		MAE	NLPD	MAE	NLPD	MAE	NLPD	MAE	NLPD	MAE	NLPD	MAE	NLPD	MAE	NLPD	MAE	NLPD
IMC	ADAM	0.619	**1.142**	0.996	**1.108**	**0.072**	**-0.940**	1.622	9.585	0.136	**-0.232**	1.407	3.282	0.687	1.418	0.718	1.365
IMC	NGD	0.619	1.144	3.594	1.174	**0.072**	-0.916	1.652	10.231	0.137	-0.189	1.604	4.293	0.671	1.488	**0.717**	1.449
LMC	ADAM	**0.616**	1.152	2.330	1.166	**0.072**	**-0.940**	1.605	9.022	0.136	-0.229	1.455	3.556	0.699	1.449	0.718	1.389
LMC	NGD	0.627	1.169	4.013	1.206	**0.072**	-0.893	1.649	9.301	0.137	-0.029	1.617	2.995	0.669	1.481	**0.717**	1.443
FSDE-SVI	ADAM	0.799	1.212	0.892	1.167	0.133	-0.316	**1.500**	**3.932**	**0.132**	-0.217	**1.157**	**1.489**	0.762	1.417	0.719	1.274
FSDE-SVI	NGD	0.683	1.162	**0.861**	1.146	0.083	-0.763	1.657	5.360	**0.132**	-0.212	1.853	2.555	**0.353**	**0.917**	0.727	**0.969**

els, we use minibatches of size $T_b = 500$ and $M = 500$ inducing points.

ENERGY. We use the data from a study for predicting energy consumption of home appliances in a low-energy building (Candanedo et al., 2017). The dataset consists of temperature measurements in $P = 10$ different rooms across the building taken at 10-minute intervals for 4.5 months, amounting to $T = 19,735$ samples. We treat the relative time of each measurement as an input and the temperature of each location as an output. We hold out the last 200 samples for extrapolation and randomly take a 80-20 split to obtain the training and interpolation data. For all models, we use minibatches of size $T_b = 1,000$ and $M = 1,000$ inducing points.

AIR QUALITY. We predict the hourly air pollutant measurements collected from the Gucheng subdistrict in Beijing between March 1st, 2013 and February 28th, 2017 (Zhang et al., 2017). We consider $P = 7$ real-valued measurements—the $PM_{2.5}, PM_{10}, SO_2, NO_2, CO, O_3$ concentration levels and temperature. We obtain measurements at $T = 26,034$ time points and standardize the data. We hold out the last 300 samples for extrapolation and randomly take a 80-20 split to obtain the training and interpolation data. For all models, we use minibatches of size $T_b = 1,000$ and $M = 2,000$ inducing points.

The results on runtime for all sparse variational inference methods on the four real datasets are summarized in Figure 4. The total runtimes in Figure 4(a) and the ELBO plots in Figure S2 of the Supplementary Material show that the factorial SDE with sparse variational inference is the only model that either reaches convergence or reaches the maximum set of iterations in under 24 hours for all real datasets. For the STOCK, ENERGY, and AIR QUALITY datasets, for which we use moderate to large numbers of inducing points, we observe significant improvements in the average time consumed per gradient update compared to the IMC and LMC baselines (Fig. 4(b)). For the COVID-19 data, the total runtime and average time per step are similar across all methods, as all methods scale linearly with respect to the number of outputs (see Table 1) and only $M = 50$ inducing points are used.

Table 4 shows that the factorial SDE achieves competitive results in both MAE and NLPD across all datasets. In particular, for the AIR QUALITY data, which has the largest number of samples, the factorial SDE trained with natural gradient descent reduces the MAE of the other models by half. This is in part due to the fact that the factorial SDE can be optimized for a larger number of iterations within the given time budget, owing to its scalability in the number of inducing points. We also observe that for the factorial SDE, using natural gradients consistently improves convergence, MAE, and NLPD across all datasets.

6 DISCUSSION AND CONCLUSION

We presented the factorial SDE for multi-output GP regression, and proposed a structured mean-field variational inference strategy that exploits the factorial structure. In both simulated and real data experiments, we showed that our approach significantly improves the scalability while achieving comparable or better prediction accuracy, when compared to existing sparse variational inference methods for the LMC and IMC.

There are several limitations of our work that remain as future work. As we only introduce inducing points and perform minibatching along the time dimension for sparse variational inference, the scalability of our proposed approach to datasets with a large number of outputs (e.g., gene expression data with a large number of genes) can be improved further by introducing additional approximations along the output dimension. In practice, the block-banded matrix operations involving the variational precision matrix can become numerically unstable during training, and employing more numerically stable approaches to the factorial SDE can further improve its performance.

Acknowledgements

This work was supported by NIH-1R21HG011116, NIH-1R21HG010948, and NSF-DBI2154089. DPJ was supported by the ARCS Foundation.

References

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: A System for Large-Scale Machine Learning. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

V. Adam, S. Eleftheriadis, A. Artemev, N. Durrande, and J. Hensman. Doubly Sparse Variational Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

M. Álvarez, D. Luengo, M. Titsias, and N. D. Lawrence. Efficient Multioutput Gaussian Processes through Variational Inducing Kernels. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

M. A. Álvarez and N. D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, 12(41):1459–1500, 2011.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. *Kernels for Vector-Valued Functions: A Review*, volume 4(3). Foundations and Trends in Machine Learning, 2012.

A. Asif and J. Moura. Block Matrices with L-Block-Banded Inverse: Inversion Algorithms. *IEEE Transactions on Signal Processing*, 53(2):630–642, 2005.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-Task Gaussian Process Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. Deisenroth, and N. Durrande. Matérn Gaussian Processes on Graphs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: Composable Transformations of Python+NumPy Programs, 2018.

L. M. Candanedo, V. Feldheim, and D. Deramaix. Data Driven Prediction Models of Energy Use of Appliances in a Low-Energy House. *Energy and Buildings*, 140:81–97, 2017.

T. Cao, J. Hall, and R. van de Geijn. Parallel Cholesky Factorization of a Block Tridiagonal Matrix. In *International Conference on Parallel Processing Workshop*, pages 327–335, 2002.

L.-F. Cheng, B. Dumitrascu, G. Darnell, C. Chivers, M. Draugelis, K. Li, and B. E. Engelhardt. Sparse Multi-Output Gaussian Processes for Online Medical Time Series Prediction. *BMC Medical Informatics and Decision Making*, 20(152), 2020.

J. Cuevas, J. Crossa, O. A. Montesinos-López, J. Burgueño, P. Pérez-Rodríguez, and G. de los Campos. Bayesian Genomic Prediction with Genotype × Environment Interaction Kernel Models. *G3 Genes—Genomes—Genetics*, 7 (1):41–53, 2017.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

E. Dong, H. Du, and L. Gardner. An Interactive Web-Based Dashboard to Track COVID-19 in Real Time. *The Lancet Infectious Diseases*, 20, 2020.

N. Durrande, V. Adam, L. Bordeaux, S. Eleftheriadis, and J. Hensman. Banded Matrix Operators for Gaussian Markov Models in the Automatic Differentiation Era. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

D. K. Duvenaud, H. Nickisch, and C. Rasmussen. Additive Gaussian Processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

J. Futoma, S. Hariharan, and K. Heller. Learning to Detect Sepsis with a Multitask Gaussian Process RNN Classifier. In *International Conference on Machine Learning (ICML)*, 2017.

Z. Ghahramani and M. I. Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29(2–3):245–273, 1997.

M. Ghassemi, M. A. F. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, and M. Feng. A Multivariate Timeseries Modeling Approach to Severity of Illness Assessment and Forecasting in ICU with Sparse, Heterogeneous Clinical Data. In *AAAI Conference on Artificial Intelligence*, 2015.

T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor and Francis, 2000.

P. Goovaerts. *Geostatistics for Natural Resources Evaluation*. Applied Geostatistics. Oxford University Press, 1997.

M. Goulard and M. Voltz. Linear Coregionalization Model: Tools for Estimation and Choice of Cross-Variogram Matrix. *Mathematical Geology*, 24(3):269–286, 1992.

A. Grigorievskiy, N. Lawrence, and S. Särkkä. Parallelizable Sparse Inverse Formulation Gaussian Processes (SpInGP). In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017.

J. Hartikainen and S. Särkkä. Kalman Filtering and Smoothing Solutions to Temporal Gaussian Process Re-

gression Models. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable Variational Gaussian Process Classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research (JMLR)*, 14(40):1303–1347, 2013.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

R. I. Kondor and J. D. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *International Conference on Machine Learning (ICML)*, 2002.

M. Lázaro-Gredilla and A. R. Figueiras-Vidal. Inter-Domain Gaussian Processes for Sparse Inference Using Inducing Features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009.

D. C. Liu and J. Nocedal. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 45(1–3):503–528, 1989.

X. Lu, A. Boukouvalas, and J. Hensman. Additive Gaussian Processes Revisited. In *International Conference on Machine Learning (ICML)*, 2022.

A. Marquand, M. Brammer, S. Williams, and O. Doyle. Bayesian Multi-Task Learning for Decoding Multi-Subject Neuroimaging Data. *Neuroimage*, 92:298 – 311, 2014.

A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian Process Library Using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.

P. Moreno-Muñoz, A. Artés-Rodríguez, and M. A. Álvarez. Heterogeneous Multi-Output Gaussian Process Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

T. V. Nguyen and E. V. Bonilla. Collaborative Multi-Output Gaussian Processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

A. V. Nikitin, S. John, A. Solin, and S. Kaski. Non-separable Spatio-Temporal Graph Kernels via SPDEs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

R. Pascanu, T. Mikolov, and Y. Bengio. On the Difficulty of Training Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*, 2013.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

M. G. Reuter and J. C. Hill. An Efficient, Block-by-Block Algorithm for Inverting a Block Tridiagonal, Nearly Block Toeplitz Matrix. *Computational Science & Discovery*, 5(1):014009, 2012.

H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 1951.

V. Rutten, A. Bernacchia, M. Sahani, and G. Hennequin. Non-Reversible Gaussian Processes for Identifying Latent Dynamical Structure in Neural Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

H. Salimbeni, S. Eleftheriadis, and J. Hensman. Natural Gradients in Practice: Non-Conjugate Variational Inference in Gaussian Process Models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

D. K. Salkuyeh. Comments on "A Note on a Three-Term Recurrence for a Tridiagonal Matrix". *Applied Mathematics and Computation*, 176(2):442–444, 2006.

L. Saul and M. Jordan. Exploiting Tractable Substructures in Intractable Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1995.

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

S. Särkkä and J. Hartikainen. Infinite-Dimensional Kalman Filtering Approach to Spatio-Temporal Gaussian Process Regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal Learning via Infinite-Dimensional Bayesian Filtering and Smoothing: A Look at Gaussian Process Regression Through Kalman Filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

M. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

M. van der Wilk, V. Dutordoir, S. John, A. Artemev, V. Adam, and J. Hensman. A Framework for Interdomain and Multioutput Gaussian Processes. *arXiv:2003.01115*, 2020.

Y. Wang and R. Khardon. Sparse Gaussian Processes for Multi-task Learning. *Machine Learning and Knowledge Discovery in Databases*, page 711–727, 2012.

J. H. Yoon, D. P. Jeong, and S. Kim. Doubly Mixed-Effects Gaussian Process Regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen. Cautionary Tales on Air-Quality Improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473, 2017.

B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, New York, 2003.

A SPARSE VARIATIONAL INFERENCE

In this section, we provide details on the factorial SDE with sparse variational inference discussed in Section 4 of the main text. Let $\{\boldsymbol{u}_\ell^{(1:M)}\}_{\ell=1}^L$ be the inducing states for the SDE representation of latent GPs $\{g_\ell(t)\}_{\ell=1}^L$ at M inducing time points $\boldsymbol{s} = [s^{(1)}, \dots, s^{(M)}]^T$. We assume that the inducing time points are identical for all $\ell \in [L]$ for simplicity, but this can be relaxed. As the state vectors in the factorial SDE representation correspond to the derivatives of $g_\ell(t)$, the inducing states can be interpreted as inter-domain inducing features (Lázaro-Gredilla and Figueiras-Vidal, 2009; Álvarez et al., 2010; Adam et al., 2020). Following standard practice (Titsias, 2009), we approximate the augmented posterior distribution $p\big(\boldsymbol{h}_{1:L}^{(1:T)}, \boldsymbol{u}_{1:L}^{(1:M)} | \boldsymbol{y}^{(1:T)}\big)$ with the variational distribution

$$q\big(\boldsymbol{h}_{1:L}^{(1:T)}, \boldsymbol{u}_{1:L}^{(1:M)}\big) = \prod_{\ell=1}^{L} p\big(\boldsymbol{h}_\ell^{(1:T)} | \boldsymbol{u}_\ell^{(1:M)}\big) q\big(\boldsymbol{u}_\ell^{(1:M)}\big),$$

where

$$q\big(\boldsymbol{u}_\ell^{(1:M)}\big) = \mathcal{N}(\boldsymbol{m}_\ell, \boldsymbol{S}_\ell^{-1}).$$

$\boldsymbol{m}_\ell \in \mathbb{R}^{MD}$ above is the variational mean and $\boldsymbol{S}_\ell$ is the $MD \times MD$ variational precision matrix with symmetric block-tridiagonal structure. Given a T_b-sized minibatch $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(T_b)}] \in \mathbb{R}^{P \times T_b}$, the ELBO for the sparse variational inference approach is

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}} &= \sum_{i=1}^{T_b} \mathbb{E}_q\big[\log p\big(\boldsymbol{y}^{(i)} | \boldsymbol{h}_{1:L}^{(i)}\big)\big] - \sum_{\ell=1}^{L} \text{KL}\left(q\big(\boldsymbol{u}_\ell^{(1:M)}\big) \big|\big| p\big(\boldsymbol{u}_\ell^{(1:M)}\big)\right) \\
&= -\frac{T_b P}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}||\boldsymbol{Y} - \hat{\boldsymbol{Y}}||_F^2 - \frac{1}{2\sigma^2}\sum_{\ell=1}^{L}||\boldsymbol{w}_\ell||^2\left(\sum_{i=1}^{T}\big[\boldsymbol{S}_\ell^{-1^{(i)}}\big]_{1,1}\right) \\
&\quad - \frac{1}{2}\sum_{\ell=1}^{L}\left[\log\frac{|\boldsymbol{S}_\ell|}{|\boldsymbol{\Lambda}_\ell|} - MD + \boldsymbol{m}_\ell^T\boldsymbol{\Lambda}_\ell\boldsymbol{m}_\ell + \text{tr}(\boldsymbol{\Lambda}_\ell\boldsymbol{S}_\ell^{-1})\right],
\end{aligned}$$

where $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}^{(1)}, \dots, \hat{\boldsymbol{y}}^{(T_b)}]$, $\hat{\boldsymbol{y}}^{(i)} = \sum_{\ell=1}^{L} \boldsymbol{w}_\ell \boldsymbol{U} \boldsymbol{m}_\ell^{(i)}$ is the prediction for the i-th sample, $\boldsymbol{\Lambda}_\ell$ is the $MD \times MD$ precision matrix of the prior $p\big(\boldsymbol{u}_\ell^{(1:M)}\big)$, and $\big(\boldsymbol{m}_\ell^{(i)}, \boldsymbol{S}_\ell^{-1^{(i)}}\big)$ are the moments of the approximate posterior for the i-th data point $q\big(\boldsymbol{h}_\ell^{(i)}\big) = \int p\big(\boldsymbol{h}_\ell^{(i)} | \boldsymbol{u}_\ell^{(1:M)}\big) q\big(\boldsymbol{u}_\ell^{(1:M)}\big) d\boldsymbol{u}_\ell^{(1:M)}$. We provide details for the forecasting and smoothing of $\big(\boldsymbol{m}_\ell^{(i)}, \boldsymbol{S}_\ell^{-1^{(i)}}\big)$ from the variational parameters $(\boldsymbol{m}_\ell, \boldsymbol{S}_\ell)$ in Section B of the Supplementary Material. Keeping the inducing states as global latent variables, we apply stochastic optimization for efficient learning and inference (Robbins and Monro, 1951; Hoffman et al., 2013; Hensman et al., 2013).

B FORECASTING AND SMOOTHING FOR PREDICTION

In this section, we provide details for computing the forecasted and smoothed moments of the approximate posterior $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ at a new time point $t^{(*)} \in \mathbb{R}$. Note that $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ takes the form

$$q\big(\boldsymbol{h}_\ell^{(*)}\big) = \begin{cases} \int p\big(\boldsymbol{h}_\ell^{(*)} | \boldsymbol{h}_\ell^{(1:T)}\big) q\big(\boldsymbol{h}_\ell^{(1:T)}\big) d\boldsymbol{h}_\ell^{(1:T)} & \text{(Factorial SDE)} \\ \int p\big(\boldsymbol{h}_\ell^{(*)} | \boldsymbol{u}_\ell^{(1:M)}\big) q\big(\boldsymbol{u}_\ell^{(1:M)}\big) d\boldsymbol{u}_\ell^{(1:M)} & \text{(Factorial SDE with sparse variational inference).} \end{cases}$$

Due to the Markov structure of the factorial representation of LMC, $\boldsymbol{h}_\ell^{(*)}$ is conditionally independent of all other states given the states in the ℓ-th SDE that are closest in time. For the factorial SDE, the closest states come from $\boldsymbol{h}_\ell^{(1:T)}$, and for the factorial SDE with sparse variational inference, the closest states come from $\boldsymbol{u}_\ell^{(1:M)}$. We make a *forecasting* prediction when $t^{(*)}$ is greater than the last training time point $t^{(T)}$ or the last inducing time point $s^{(M)}$. We make a *smoothing* prediction when $t^{(*)}$ is between two training time points $t^{(i-1)}$ and $t^{(i)}$ for some $i \in [T]$ or between two inducing time points $s^{(j-1)}$ and $s^{(j)}$ for some $j \in [M]$.

Below, we provide the expressions of the approximate moments for both the non-sparse and sparse variational inference settings. We can then plug in the resulting approximate moments into the posterior predictive distribution $q\big(\boldsymbol{y}^{(*)}\big) \approx p\big(\boldsymbol{y}^{(*)} | \boldsymbol{y}^{(1:T)}\big)$ in Section 4 of the main text to predict the outputs at unobserved time points.

B.1 Factorial SDE

For the factorial SDE, the forecasting moments of $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ are given by

$$\begin{aligned}\boldsymbol{m}_\ell^{(*)} &= e^{\Delta_* \boldsymbol{A}_\ell}\boldsymbol{m}_\ell^{(T)},\\ \boldsymbol{S}_\ell^{-1^{(*)}} &= e^{\Delta_* \boldsymbol{A}_\ell}\boldsymbol{S}_\ell^{-1^{(T)}}e^{\Delta_* \boldsymbol{A}_\ell^T} + \int_0^{\Delta_*} e^{(\Delta_*-\tau)\boldsymbol{A}_\ell}\boldsymbol{B}Q_\ell\boldsymbol{B}^T e^{(\Delta_*-\tau)\boldsymbol{A}_\ell^T}d\tau,\end{aligned}$$

where $\Delta_* \triangleq t^{(*)} - t^{(T)}$, and $(\boldsymbol{m}_\ell^{(T)}, \boldsymbol{S}_\ell^{-1^{(T)}})$ are the moments of $q\big(\boldsymbol{h}_\ell^{(T)}\big)$.

The smoothing moments of $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ are given by

$$\begin{aligned}\boldsymbol{m}_\ell^{(*)} &= \boldsymbol{M}_{\ell,1}\boldsymbol{m}_\ell^{(i-1)} + \boldsymbol{M}_{\ell,2}\boldsymbol{m}_\ell^{(i)},\\ \boldsymbol{S}_\ell^{-1^{(*)}} &= \big[\boldsymbol{M}_{\ell,1}, \boldsymbol{M}_{\ell,2}\big]\boldsymbol{S}_\ell^{-1^{(i-1:i)}}\big[\boldsymbol{M}_{\ell,1}, \boldsymbol{M}_{\ell,2}\big]^T + \boldsymbol{\Sigma}_\ell^{(*|i-1:i)},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{M}_{\ell,1} &= e^{\Delta_*^{\text{prev}}\boldsymbol{A}_\ell} + \boldsymbol{\Sigma}_\ell^{(*,i-1)}{\boldsymbol{\Psi}_\ell^{(i)}}^T{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Psi}_\ell^{(i)} - \boldsymbol{\Sigma}_\ell^{(*,i)}{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Psi}_\ell^{(i)},\\ \boldsymbol{M}_{\ell,2} &= -\boldsymbol{\Sigma}_\ell^{(*,i-1)}{\boldsymbol{\Psi}_\ell^{(i)}}^T{\boldsymbol{\Phi}_\ell^{(i)}}^{-1} + \boldsymbol{\Sigma}_\ell^{(*,i)}{\boldsymbol{\Phi}_\ell^{(i)}}^{-1},\end{aligned}$$

with $\Delta_*^{\text{prev}} \triangleq t^{(*)} - t^{(i-1)}$, $\Delta_*^{\text{next}} \triangleq t^{(i)} - t^{(*)}$, $\boldsymbol{\Sigma}_\ell^{(*,i)}$ denoting the prior cross-covariance matrix of $\boldsymbol{h}_\ell^{(*)}$ and $\boldsymbol{h}_\ell^{(i)}$, $\boldsymbol{\Sigma}_\ell^{(*|i-1:i)}$ denoting the prior conditional covariance matrix of $\boldsymbol{h}_\ell^{(*)}$ given $\boldsymbol{h}_\ell^{(i-1)}$ and $\boldsymbol{h}_\ell^{(i)}$, and $\boldsymbol{S}_\ell^{-1^{(i-1:i)}}$ denoting the $2D \times 2D$ block $[\boldsymbol{S}_\ell^{-1}]_{i-1:i,i-1:i}$. The exact expressions for the cross-covariance and conditional covariance matrices can be derived from the prior joint distribution over $\boldsymbol{h}_\ell^{(i-1)}, \boldsymbol{h}_\ell^{(*)}$, and $\boldsymbol{h}_\ell^{(i)}$ and are given as

$$\begin{aligned}\boldsymbol{\Sigma}_\ell^{(*,i-1)} &= e^{\Delta_*^{\text{prev}}\boldsymbol{A}_\ell}\boldsymbol{\Sigma}_{\ell,\infty},\\ \boldsymbol{\Sigma}_\ell^{(*,i)} &= \boldsymbol{\Sigma}_{\ell,\infty}e^{\Delta_*^{\text{next}}\boldsymbol{A}_\ell^T},\\ \boldsymbol{\Sigma}_\ell^{(*|i-1:i)} &= \boldsymbol{\Sigma}_\ell^{(*)} - \boldsymbol{\Sigma}_\ell^{(*,i-1:i)}\boldsymbol{\Sigma}_\ell^{(i-1:i)^{-1}}\boldsymbol{\Sigma}_\ell^{(*,i-1:i)^T}\\ &= e^{\Delta_*\boldsymbol{A}_\ell} + \boldsymbol{\Sigma}_\ell^{(*,i)}{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Psi}_\ell^{(i)}\boldsymbol{\Sigma}_\ell^{(*,i-1)^T} + \boldsymbol{\Sigma}_\ell^{(*,i-1)}{\boldsymbol{\Psi}_\ell^{(i)}}^T{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Sigma}_\ell^{(*,i)^T}\\ &\quad - \boldsymbol{\Sigma}_\ell^{(*,i-1)}{\boldsymbol{\Psi}_\ell^{(i)}}^T{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Psi}_\ell^{(i)}\boldsymbol{\Sigma}_\ell^{(*,i-1)^T} - \boldsymbol{\Sigma}_\ell^{(*,i)}{\boldsymbol{\Phi}_\ell^{(i)}}^{-1}\boldsymbol{\Sigma}_\ell^{(*,i)^T}.\end{aligned}$$

B.2 Factorial SDE with Sparse Variational Inference

For the factorial SDE with sparse variational inference, the forecasting moments of $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ are given by

$$\begin{aligned}\boldsymbol{m}_\ell^{(*)} &= e^{\Delta_* \boldsymbol{A}_\ell}\boldsymbol{m}_\ell^{(M)},\\ \boldsymbol{S}_\ell^{-1^{(*)}} &= e^{\Delta_* \boldsymbol{A}_\ell}\boldsymbol{S}_\ell^{-1^{(M)}}e^{\Delta_* \boldsymbol{A}_\ell^T} + \int_0^{\Delta_*} e^{(\Delta_*-\tau)\boldsymbol{A}_\ell}\boldsymbol{B}Q_\ell\boldsymbol{B}^T e^{(\Delta_*-\tau)\boldsymbol{A}_\ell^T}d\tau,\end{aligned}$$

where $\Delta_* \triangleq t^{(*)} - s^{(M)}$, and $(\boldsymbol{m}_\ell^{(M)}, \boldsymbol{S}_\ell^{-1^{(M)}})$ are the moments of $q\big(\boldsymbol{u}_\ell^{(M)}\big)$.

The smoothing moments of $q\big(\boldsymbol{h}_\ell^{(*)}\big)$ are given by

$$\begin{aligned}\boldsymbol{m}_\ell^{(*)} &= \boldsymbol{M}_{\ell,1}\boldsymbol{m}_\ell^{(j-1)} + \boldsymbol{M}_{\ell,2}\boldsymbol{m}_\ell^{(j)},\\ \boldsymbol{S}_\ell^{-1^{(*)}} &= \big[\boldsymbol{M}_{\ell,1}, \boldsymbol{M}_{\ell,2}\big]\boldsymbol{S}_\ell^{-1^{(j-1:j)}}\big[\boldsymbol{M}_{\ell,1}, \boldsymbol{M}_{\ell,2}\big]^T + \boldsymbol{\Sigma}_\ell^{(*|j-1:j)},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{M}_{\ell,1} &= e^{\Delta_*^{\text{prev}}\boldsymbol{A}_\ell} + \boldsymbol{\Sigma}_\ell^{(*,j-1)}{\boldsymbol{\Psi}_\ell^{(j)}}^T{\boldsymbol{\Phi}_\ell^{(j)}}^{-1}\boldsymbol{\Psi}_\ell^{(j)} - \boldsymbol{\Sigma}_\ell^{(*,j)}{\boldsymbol{\Phi}_\ell^{(j)}}^{-1}\boldsymbol{\Psi}_\ell^{(j)},\\ \boldsymbol{M}_{\ell,2} &= -\boldsymbol{\Sigma}_\ell^{(*,j-1)}{\boldsymbol{\Psi}_\ell^{(j)}}^T{\boldsymbol{\Phi}_\ell^{(j)}}^{-1} + \boldsymbol{\Sigma}_\ell^{(*,j)}{\boldsymbol{\Phi}_\ell^{(j)}}^{-1}.\end{aligned}$$

with $\Delta_*^{\text{prev}} \triangleq t^{(*)} - s^{(j-1)}, \Delta_*^{\text{next}} \triangleq s^{(j)} - t^{(*)}$, $\boldsymbol{\Sigma}_\ell^{(*,j)}$ denoting the prior cross-covariance matrix of $\boldsymbol{h}_\ell^{(*)}$ and $\boldsymbol{u}_\ell^{(j)}$, $\boldsymbol{\Sigma}_\ell^{(*|j-1:j)}$ denoting the prior conditional covariance matrix of $\boldsymbol{h}_\ell^{(*)}$ given $\boldsymbol{u}_\ell^{(j-1)}$ and $\boldsymbol{u}_\ell^{(j)}$, and $\boldsymbol{S}_\ell^{-1^{(j-1:j)}}$ denoting the $2D \times 2D$ block $[\boldsymbol{S}_\ell^{-1}]_{j-1:j,j-1:j}$. As in the smoothing equations for the factorial SDE, the exact expressions for the cross-covariance and conditional covariance matrices can be derived from the prior joint distribution over $\boldsymbol{u}_\ell^{(j-1)}, \boldsymbol{h}_\ell^{(*)}$, and $\boldsymbol{u}_\ell^{(j)}$ and are given as

$$
\begin{aligned}
\boldsymbol{\Sigma}_\ell^{(*,j-1)} &= e^{\Delta_*^{\text{prev}} \boldsymbol{A}_\ell} \boldsymbol{\Sigma}_{\ell,\infty}, \\
\boldsymbol{\Sigma}_\ell^{(*,j)} &= \boldsymbol{\Sigma}_{\ell,\infty} e^{\Delta_*^{\text{next}} \boldsymbol{A}_\ell^T}, \\
\boldsymbol{\Sigma}_\ell^{(*|j-1:j)} &= \boldsymbol{\Sigma}_\ell^{(*)} - \boldsymbol{\Sigma}_\ell^{(*,j-1:j)} \boldsymbol{\Sigma}_\ell^{(j-1:j)^{-1}} \boldsymbol{\Sigma}_\ell^{(*,j-1:j)^T} \\
&= e^{\Delta_* \boldsymbol{A}_\ell} + \boldsymbol{\Sigma}_\ell^{(*,j)} \boldsymbol{\Phi}_\ell^{(j)^{-1}} \boldsymbol{\Psi}_\ell^{(j)} \boldsymbol{\Sigma}_\ell^{(*,j-1)^T} + \boldsymbol{\Sigma}_\ell^{(*,j-1)} \boldsymbol{\Psi}_\ell^{(j)^T} \boldsymbol{\Phi}_\ell^{(j)^{-1}} \boldsymbol{\Sigma}_\ell^{(*,j)^T} \\
&\quad - \boldsymbol{\Sigma}_\ell^{(*,j-1)} \boldsymbol{\Psi}_\ell^{(j)^T} \boldsymbol{\Phi}_\ell^{(j)^{-1}} \boldsymbol{\Psi}_\ell^{(j)} \boldsymbol{\Sigma}_\ell^{(*,j-1)^T} - \boldsymbol{\Sigma}_\ell^{(*,j)} \boldsymbol{\Phi}_\ell^{(j)^{-1}} \boldsymbol{\Sigma}_\ell^{(*,i)^T}.
\end{aligned}
$$

C ADDITIONAL EXPERIMENTAL DETAILS

In this section, we provide additional results and details on the experimental setting for the simulation and real data experiments.

C.1 Convergence on LARGE-SIM Data

Figure S1 shows the ELBO over time and over iterations for all methods on one train-test split of the LARGE-SIM data, as we vary the number of inducing points M. The optimization for the factorial SDE with sparse variational inference converges faster in terms of both time and iterations across all settings of M, compared to the IMC and LMC baselines. For the IMC and LMC baselines, as M increases, optimization often fails to converge within 24 hours, due to their cubic dependence on M (the baselines with natural gradient descent in Figs. S1(b) and S1(c) and all baselines in Figs. S1(d) and S1(e)). In addition, we see that using natural gradients for the factorial SDE often speeds up convergence, especially when M is large.

C.2 Additional Details on Real Data Experiments

COVID-19 For all models, we use 300 latent GPs, to accommodate the large number of outputs $P = 3,019$. When Adam is used to optimize both the kernel hyperparameters and the variational parameters, we set $\eta_1 = \eta_2 = 10^{-3}$ for the IMC and LMC baselines, and $\eta_1 = 10^{-3}$ and $\eta_2 = 10^{-4}$ for the factorial SDE with sparse variational inference. When using natural gradients for the variational parameters, we set $\eta_2 = 10^{-2}$ for the IMC and LMC baselines, and change η_2 from 10^{-5} to 10^{-4} over 15,000 iterations for the factorial SDE with sparse variational inference.

STOCK For all models, we use 15 latent GPs. When Adam is used to optimize both the kernel hyperparameters and the variational parameters, we set $\eta_1 = \eta_2 = 10^{-3}$ for all models. When using natural gradients for the variational parameters, we set $\eta_2 = 10^{-2}$ for the IMC and LMC baselines, and change η_2 from 10^{-5} to 10^{-3} over 1,000 iterations for the factorial SDE with sparse variational inference.

ENERGY For all models, we use 5 latent GPs. When Adam is used to optimize both the kernél hyperparameters and the variational parameters, we set $\eta_1 = \eta_2 = 10^{-3}$ for all models. When using natural gradients for the variational parameters, we set $\eta_2 = 10^{-2}$ for the IMC and LMC baselines, and change η_2 from 10^{-5} to 10^{-3} over 1,000 iterations for the factorial SDE with sparse variational inference.

AIR QUALITY For all models, we use 5 latent GPs. When Adam is used to optimize both the kernel hyperparameters and the variational parameters, we set $\eta_1 = \eta_2 = 10^{-3}$ for all models. When using natural gradients for the variational parameters, we set $\eta_2 = 10^{-2}$ for the IMC and LMC baselines, and change η_2 from 10^{-5} to 10^{-3} over 1,000 iterations for the factorial SDE with sparse variational inference.

C.3 Convergence on Real Data

Figure S2 shows the ELBO over time and over iterations for all methods on the four real-world datasets. For the relatively small COVID-19 dataset with a relatively small number of inducing points $M = 50$, all methods have similar performance: all baseline methods with natural gradient descent converge, and the other baselines and the factorial SDE with sparse variational inference proceed until the maximum 50,000 iterations is reached within 24 hours (Fig. S2(a)). However, for larger datasets with more inducing points such as the STOCK ($M = 500$), ENERGY ($M = 1,000$), and AIR QUALITY ($M = 2,000$) datasets, the factorial SDE significantly outperforms the baseline methods. The optimization of all of the baselines fails to converge within 24 hours, whereas optimization of the factorial SDE reaches approximate convergence (i.e., the ELBO plateaus) in significantly less time (Figs. S2(b)-S2(d)). As in the simulation experiments with the LARGE-SIM data, these results empirically demonstrate that the linear dependence on M for the factorial SDE with sparse variational inference results in significantly faster learning compared to the IMC and LMC baselines with cubic dependence on M. We also see that using natural gradients for the factorial SDE consistently achieves higher ELBO values across all datasets.

D SOFTWARE

The software is available at `https://github.com/SeyoungKimLab/FactorialSDE`.

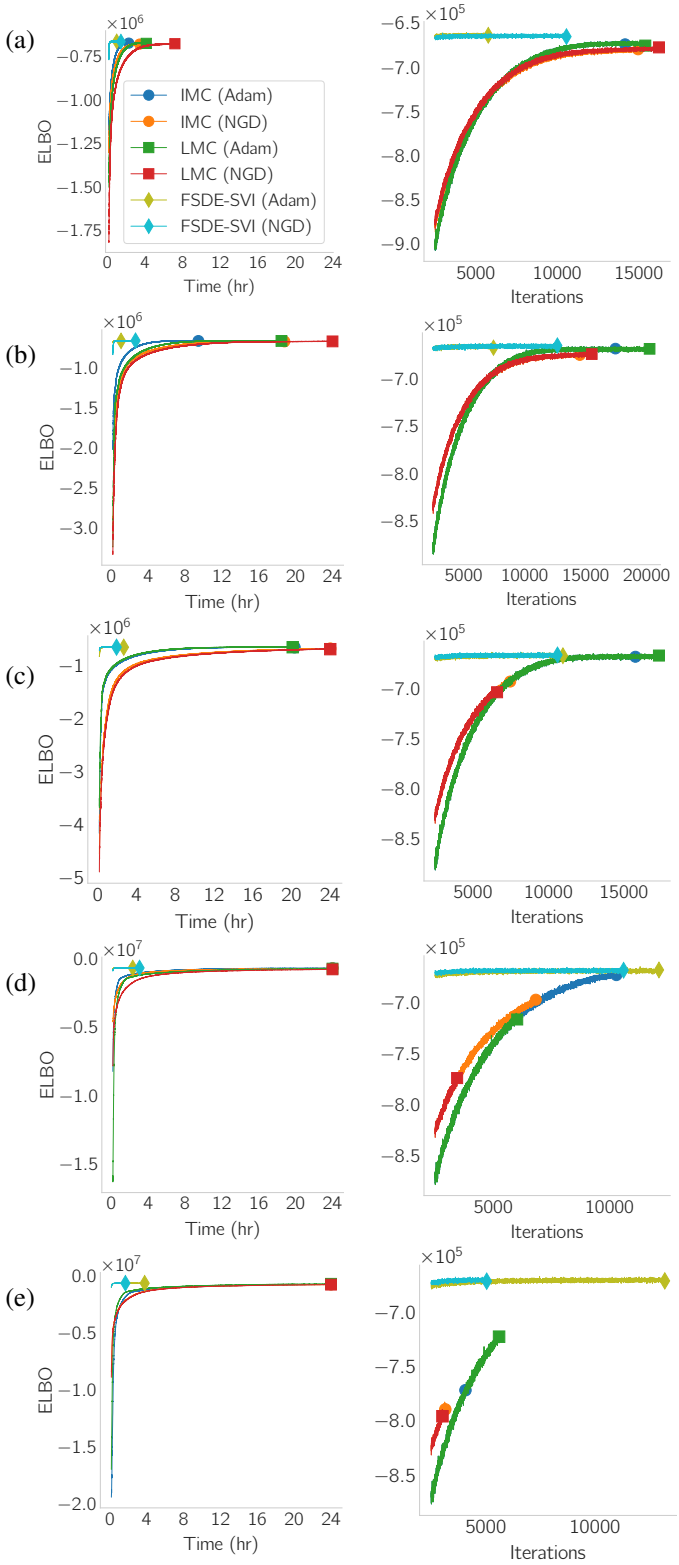


Figure S1: Convergence of different sparse variational inference methods on the LARGE-SIM data. The ELBO over time after the first 10 minutes (left column) and over iterations after the first 2,500 iterations (right column) are shown for different numbers of inducing points M. (a) $M = 200$, (b) $M = 400$, (c) $M = 600$, (d) $M = 800$, and (e) $M = 1,000$.

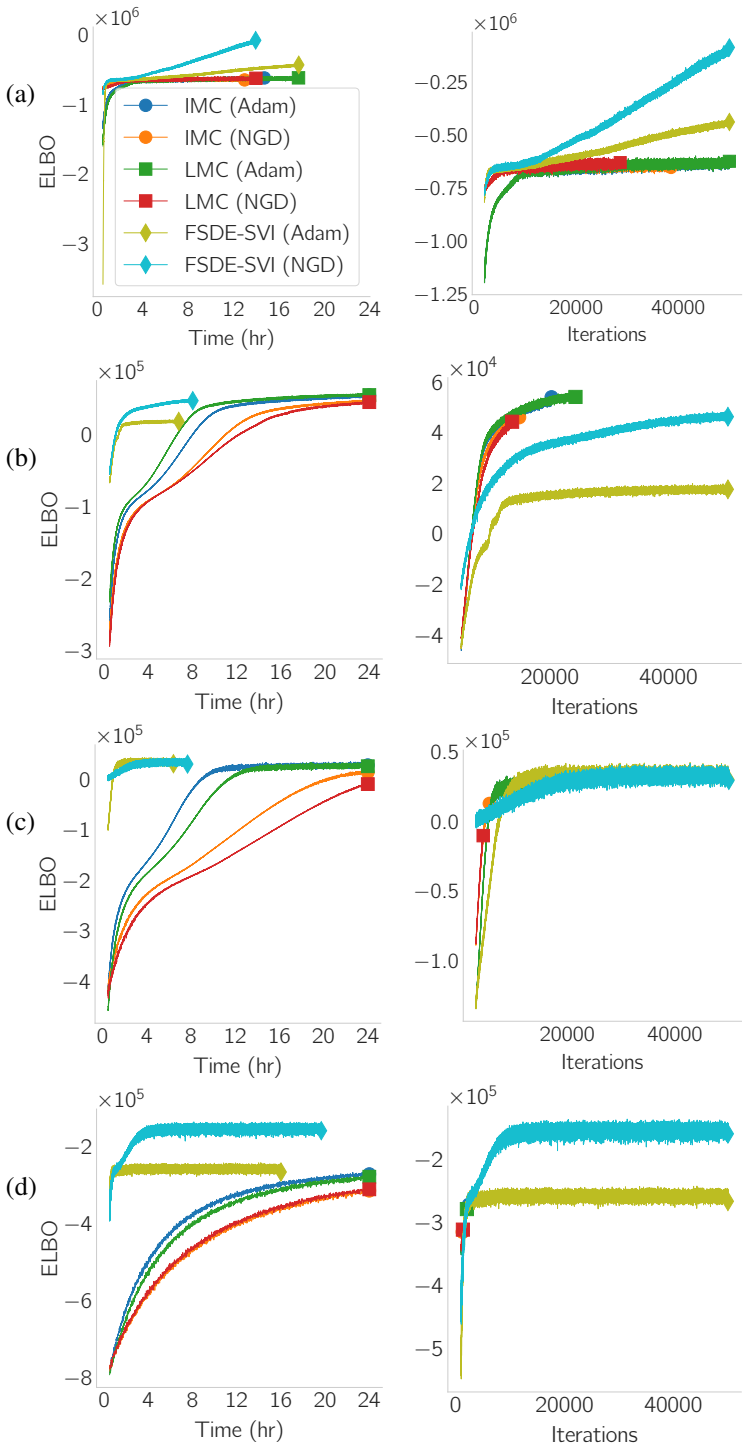


Figure S2: Convergence of different sparse variational inference methods on the four real datasets. (a) COVID-19, (b) STOCK, (c) ENERGY, and (d) AIR QUALITY. The left column plots the ELBO over time after the first 10 minutes, and the right column plots the ELBO over iterations after the first 2,000 iterations for COVID-19, 5,000 iterations for STOCK, 3,000 iterations for ENERGY, and 1,000 iterations for AIR QUALITY.