Traversing Time with Multi-Resolution Gaussian Process State-Space Models

Krista Longi KRISTA.LONGI@HELSINKI.FI
Jakob Lindinger JAKOB.LINDINGER@DE.BOSCH.COM
Olaf Duennbier OLAF.DUENNBIER@DE.BOSCH.COM
Melih Kandemir KANDEMIR@IMADA.SDU.DK
Arto Klami ARTO.KLAMI@HELSINKI.FI
Barbara Rakitsch BARBARA.RAKITSCH@DE.BOSCH.COM

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

Gaussian Process state-space models capture complex temporal dependencies in a principled manner by placing a Gaussian Process prior on the transition function. These models have a natural interpretation as discretized stochastic differential equations, but inference for long sequences with fast and slow transitions is difficult. Fast transitions need tight discretizations whereas slow transitions require backpropagating the gradients over long subtrajectories. We propose a novel Gaussian process state-space architecture composed of multiple components, each trained on a different resolution, to model effects on different timescales. The combined model allows traversing time on adaptive scales, providing efficient inference for arbitrarily long sequences with complex dynamics. We benchmark our novel method on semi-synthetic data and on an engine modeling task. Both experiments show that our approach compares favorably against its state-of-the-art alternatives.

Keywords: State-Space Models, Gaussian Processes, System Identification

1. Introduction

System identification refers to learning dynamical systems from data and lies at the heart of many control applications such as epidemic forecasting (Zimmer and Yaesoubi, 2020) for public health policies, reinforcement learning for portfolio management (Heaton et al., 2017), or emission modeling (Yu et al., 2020) for calibrating the car engine. In many cases, we do not know the underlying physical model but instead need to learn the dynamics from data only, ideally in a non-parametric manner to support arbitrary dynamics. Irrespective of the total amount of data, many interesting phenomena manifest only in a small subset of the samples, which does not allow to uniquely identify the underlying dynamics and, instead, call for probabilistic methods.

Gaussian Process state-space models (GPSSMs) hold the promise to model non-linear, unknown dynamics in a probabilistic manner by placing a Gaussian Process (GP) prior on the transition function (Wang et al., 2005; Frigola, 2015). While inference has been proven to be challenging for this model family, there has been a lot of progress in the past years and recent approaches vastly improved the scalability (Eleftheriadis et al., 2017; Doerr et al., 2018).

For long trajectories, methods updating the parameters using the complete sequence converge poorly due to the vanishing and exploding gradient problem (Pascanu et al., 2013). While special-

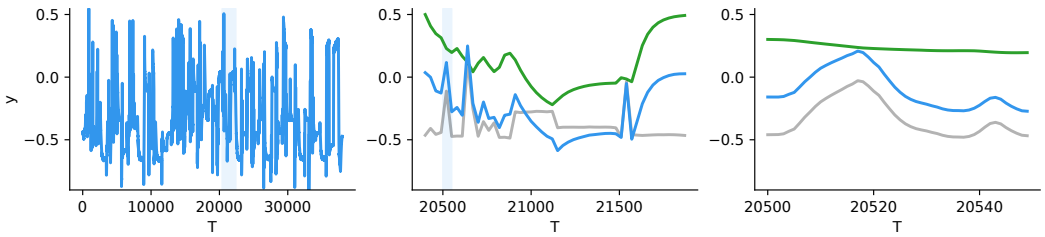

Figure 1: **Resolution Matters.** A semi-synthetic dataset (blue) created as a sum of two functions, one with fast varying dynamics corresponding to large updates (gray) and one with slowly varying dynamics corresponding to small updates between adjacent states (green). *Left:* Shown is the complete trajectory. Using all observations for each parameter update is too time- and memory-consuming. *Middle:* Shown is a dilated mini-batch of size 50 for which we have selected every 30th observation. It allows for fitting the slowly varying function (green). The fast dynamics (gray) cannot be inferred as the observations are too sparse. *Right:* Shown is a mini-batch of size 50 on the standard resolution. The mini-batch only covers a short interval of the trajectory as can be noted by the different time axis. It allows for fitting the fast varying function (gray). The slow dynamics (green) cannot be inferred as the gradient information is too weak. Our algorithm allows learning on multiple resolutions to capture effects on different timescales.

ized architectures help circumventing the problem in the case of recurrent neural networks (Hochreiter and Schmidhuber, 1997; Chung et al., 2014), it is not clear how one can apply these concepts to GP models. Furthermore, the problem of large runtime and memory footprints for training persists, as the gradients need to be backpropagated through the complete sequence. A natural solution is to divide the trajectory into mini-batches which reduces training time significantly, but also lowers the flexibility of the model: long-term effects that evolve slower than the size of one mini-batch can no longer be inferred (Williams and Zipser, 1995).

To address the problem of modeling long-term dependencies while retaining the computational advantage of mini-batching, we propose a novel GPSSM architecture with L additive components. The resulting posterior is intractable, and we apply variational inference to find an efficient and structured approximation (Blei et al., 2017). To capture effects on different time scales, our training scheme cycles through the components, whereby each component is trained on a different resolution. For training the low-resolution components, we downsample the observations of the sequence, allowing us to pack a longer history in a mini-batch of fixed size (see Figure 1). Our training algorithm is grounded in a coherent statistical framework by interpreting the GP transition model as a stochastic differential equation (SDE) similar to Hegde et al. (2019). This allows us to train each component with a different resolution under a unifying framework.

We validate our new algorithm experimentally and show that it works well in practice on semi-synthetic data and on a challenging engine modeling task. Furthermore, we demonstrate that our algorithm outperforms its competitors by a large margin in cases where the dataset consists of fast and slow dynamics. For the engine modeling task, we introduce a new dataset to the community that contains the raw emissions of a gasoline car engine and has over 500,000 measurements. The dataset is available at https://github.com/boschresearch/Bosch-Engine-Datasets.

2. Background on GPSSMs and SDEs

Gaussian Processes in a Nutshell The GP prior, $f(x) \sim GP(0, k(x, x'))$, defines a distribution over functions, $f : \mathbb{R}^{D_x} \to \mathbb{R}$, and is fully specified by the kernel $k : \mathbb{R}^{D_x} \times \mathbb{R}^{D_x} \to \mathbb{R}$. Given a set of arbitrary inputs, $x_M = \{x_m\}_{m=1}^M$, their function values, $f_M = \{f(x_m)\}_{m=1}^M$, follow a Gaussian distribution $p(f_M) = \mathcal{N}(f_M|0, \mathrm{K}_{MM})$ where $\mathrm{K}_{MM} = \{k(x_m, x_{m'})\}_{m=1,m'=1}^M$.

For a new set of input points, $x_N = \{x_n\}_{n=1}^N$, the predictive distribution over the corresponding function values, $f_N = \{f(x_n)\}_{n=1}^N$, can then be obtained by conditioning the joint distribution on f_M, leading to $p(f_N|f_M) = \mathcal{N}(f_N|\mu(x_N), \Sigma(x_N))$ with

$$\mu(x_N) = \mathrm{K}_{NM}\mathrm{K}_{MM}^{-1} f_M, \quad \Sigma(x_N) = \mathrm{K}_{NN} - \mathrm{K}_{NM}\mathrm{K}_{MM}^{-1}\mathrm{K}_{NM}^\top, \tag{1}$$

where the cross-covariances K_{NM} are defined similarly as K_{MM}, i.e. $\mathrm{K}_{NM} = \{k(x_n, x_m)\}_{n,m=1}^{N,M}$. For a more detailed introduction, we refer the interested reader to Rasmussen and Williams (2006).

Gaussian Process State-Space Models We are given a dataset $y_{1:T} = \{y_t\}_{t=1}^T$ over T time points, where $y_t \in \mathbb{R}^{D_y}$ denotes the output at time point t. State-space models (see e.g. Särkkä, 2013) offer a general way to describe time-series data by introducing a latent state, $x_t \in \mathbb{R}^{D_x}$, that captures the compressed history of the system, for each time point $t \in \{1, \dots, T\}$. Assuming the process and observational noise to be i.i.d. Gaussian distributed, the model can be written down as follows:

$$x_{t+1}|x_t \sim \mathcal{N}(x_{t+1}|x_t + f(x_t), \mathrm{Q}), \qquad y_t|x_t \sim \mathcal{N}(y_t|g(x_t), \Omega),$$

where $f : \mathbb{R}^{D_x} \to \mathbb{R}^{D_x}$ models the change of the latent state in time and $g : \mathbb{R}^{D_x} \to \mathbb{R}^{D_y}$ maps the latent state to the observational space. The covariance $\mathrm{Q} \in \mathbb{R}^{D_x \times D_x}$ describes the process noise, and $\Omega \in \mathbb{R}^{D_y \times D_y}$ the observational noise. Following the literature (Wang et al., 2005; Deisenroth and Rasmussen, 2011), we assume that the update in the latent state can be modeled under a GP prior, i.e. $f(x) \sim GP(0, k(x, x'))$.[1] The model generalizes easily to problems with exogenous inputs that we left out in favor of an uncluttered notation.

Finally, we chose in our experiments a linear model $g(x_t) = Cx_t$ with output matrix $C \in \mathbb{R}^{D_y \times D_x}$ as emission function. This is a widely adopted design choice, since the linear emission model reduces non-identifiabilities of the solution (Frigola, 2015). Our approach is directly applicable to non-linear parametric emission models as well, which might for instance be important in cases in which prior knowledge supports the use of more expressive emission models.

Sparse Parametric Gaussian Process State-Space Models Sparse GPs augment the model by a set of inducing points (x_M, f_M) that can be exploited during inference to summarize the training data in an efficient way. Snelson and Ghahramani (2005) introduced the so-called FITC (fully independent training conditional) approximation on the augmented joint density $p(f_M, f_N)$ by assuming independence between the function values, f_N, conditioned on the set of inducing points, f_M. The FITC approximation has also been used previously for GPSSMs (Doerr et al., 2018), and we follow this line of work by assuming the same conditional factorization,

$$f_M \sim \mathcal{N}(f_M|0, \mathrm{K}_{MM}), \tag{2}$$

$$f_t|f_M \sim \mathcal{N}(f_t|\mu(x_t), \Sigma(x_t)), \tag{3}$$

$$x_{t+1}|x_t, f_t \sim \mathcal{N}(x_{t+1}|x_t + f_t, \mathrm{Q}), \tag{4}$$

1. To be more precise, each latent dimension $d \in \{1, \dots, D_x\}$ follows an independent GP prior. We suppressed the dependency on the latent dimension d for the sake of better readability in our notation.

where f_t are the GP predictions at time index t with mean $\mu(x_t)$ and covariance $\Sigma(x_t)$ [Eq. (1)]. We discuss the FITC approximation in more detail in the extended manuscript (Longi et al., 2021).

Gaussian Process Stochastic Differential Equations SDEs can be regarded as a stochastic extension to ordinary differential equations where randomness enters the system via Brownian motion. Their connection to GPSSMs is obtained by considering the SDE

$$dx_t = f(x_t)dt + \sqrt{\mathrm{Q}^{\Delta}}dW_t, \tag{5}$$

where the drift term is given by the GP predictions $f(x_t) \sim \mathcal{N}(\mu^{\Delta}(x_t), \Sigma^{\Delta}(x_t))$ [Eq. (1)], the diffusion term by $\sqrt{\mathrm{Q}^{\Delta}}$, and the Brownian motion by $W_t \in \mathbb{R}^{D_x}$. In order to clearly distinguish the notation from the canonical GPSSM [Eqs. (2) - (4)], we endow all potentially different quantities with a Δ. Applying a GP prior over the drift function has been done previously, for example in Ruttor et al. (2013), Yildiz et al. (2018), and Zhao et al. (2020). A related parameterization has also been suggested by Hegde et al. (2019) to extend deep GPs to an infinite number of hidden layers. Since the diffusion term is a random variable, the solution to Eq. (5) is non-deterministic and results in a stochastic process over x_t. Except for a few cases, such as linear time-invariant systems, SDEs cannot be solved analytically and require numerical integration. Hence, we apply the Euler-Maruyama scheme (see e.g. Särkkä and Solin, 2019) to draw approximate samples, using that $W_{j+1} - W_j \sim \mathcal{N}(0, R\Delta_t)$:

$$f_j | f_M \sim \mathcal{N}(f_j | \mu^{\Delta}(x_j), \Sigma^{\Delta}(x_j)), \tag{6}$$

$$x_{j+1} | x_j, f_j \sim \mathcal{N}(x_{j+1} | x_j + R\Delta_t f_j, R\Delta_t \mathrm{Q}^{\Delta}), \tag{7}$$

where f_j corresponds to the GP prediction at index j. The stepsize is given by $R\Delta_t$ where R is the resolution and Δ_t is the time interval between two adjacent observations in the time series $y_{1:T}$. Note that we employ the index j to denote the time indices in the Euler-Maruyama scheme, whereas we use the index t in the canonical GPSSM formulation. Consequently, a time index t indicates a time $t\Delta_t$ after the starting time, whereas the index j signifies a time $jR\Delta_t$ after the starting time. The Euler-Maruyama method converges to the true solution with shrinking step size $R\Delta_t$ with strong order of convergence of 1/2.

3. Multi-Resolution Gaussian Process State-Space Models

Standard training of GPSSM models is restricted to a single resolution which hampers inference for long sequences with fast and slow transitions. In this work, we introduce a novel GPSSM architecture that decomposes the latent space into multiple independent components. We first extend doubly-stochastic variational inference for this model class. Then, we show that this inference scheme can be generalized such that each component is learned with a dedicated resolution in order to capture effects on different timescales.

3.1. Probabilistic Model

Our model splits the latent state into L components, $x_t = \{x_t^{(l)}\}_{l=1}^{L}$, that evolve independently over time (see Figure 2):

$$f_t^{(l)} | f_M^{(l)} \sim \mathcal{N}(f_t^{(l)} | \mu^{(l)}(x_t^{(l)}), \Sigma^{(l)}(x_t^{(l)})) \tag{8}$$

$$x_{t+1}^{(l)} | x_t^{(l)}, f_t^{(l)} \sim \mathcal{N}(x_{t+1}^{(l)} | x_t^{(l)} + f_t^{(l)}, \mathrm{Q}^{(l)}). \tag{9}$$

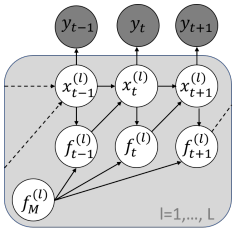

Figure 2: **Plate diagram.**

All terms are given by their equivalents in Eqs. (3) and (4) with $x_t^{(l)} \in \mathbb{R}^{D_l}$ and $\sum_l D_l = D_x$. Note that our proposed model can be cast into the standard formulation (Section 2) when allowing separate kernel hyperparameters for each latent state: the kernel hyperparameters are shared for all latent states within one component, and the latent component $x_t^{(l)}$ depends only on $x_{t-1}^{(l)}$ by the use of automatic relevance determination. The structured latent space enables learning each component with a different resolution (see Section 3.3) which would not be possible within the standard framework.

It is also worth noting that the latent states $x_t^{(1)}, \dots x_t^{(L)}$ are coupled over the emission model only. Without this coupling, the proposed model would reduce to L independent state-space models that could be trained in isolation, each on a different resolution.

Augmented model Collecting and simplifying all terms, we arrive at the augmented joint density

$$p(x_{0:T}, f_M, y_{1:T}) = \prod_{l=1}^{L} p(x_0^{(l)}) \prod_{l=1}^{L} p(f_M^{(l)}) \prod_{t=1}^{T} p(y_t|x_t) \prod_{t=0,l=1}^{T-1,L} p(x_{t+1}^{(l)}|x_t^{(l)}, f_M^{(l)}), \tag{10}$$

where $x_0 = \{x_0^{(l)}\}_{l=1}^{L}$ are the initial latent states and $x_{0:T} = \{x_t\}_{t=0}^{T}$ denote the time-series of latent states. We assume that their distribution decomposes between the components and $p(x_0^{(l)}) = \mathcal{N}(x_0^{(l)}|\mu_0^{(l)}, \mathrm{Q}_0^{(l)})$ with mean $\mu_0^{(l)} \in \mathbb{R}^{D_l}$ and covariance $\mathrm{Q}_0^{(l)} \in \mathbb{R}^{D_l \times D_l}$. The transition probability $p(x_{t+1}^{(l)}|x_t^{(l)}, f_M^{(l)}) = \mathcal{N}(x_{t+1}^{(l)}|x_t^{(l)} + \mu^{(l)}(x_t^{(l)}), \mathrm{Q}^{(l)} + \Sigma^{(l)}(x_t^{(l)}))$ is obtained by marginalizing out the $f_t^{(l)}$ [which we assume to be conditionally independent given the $f_M^{(l)}$, see Eq. (8)] from Eq. (9) via standard Gaussian integrals. While it is hard to read out from the formulas directly, analytically marginalizing out the inducing points $f_M^{(l)}$ from Eq. (10) leads to a coupling between all latent states $x_T^{(l)}$ as we show in the extended version (Longi et al., 2021).

3.2. Training over a Single Resolution

Multi-component GPSSMs can be trained over a single resolution by extending the work of Doerr et al. (2018). We start by introducing the approximate posterior,

$$q(x_{0:T}, f_M) = \prod_{l=1}^{L} q(x_0^{(l)}) \prod_{l=1}^{L} q(f_M^{(l)}) \prod_{t=0,l=1}^{T-1,L} p(x_{t+1}^{(l)}|x_t^{(l)}, f_M^{(l)}), \tag{11}$$

that decomposes across the components. Here, the conditional $p(x_{t+1}^{(l)}|x_t^{(l)}, f_M^{(l)})$ corresponds to the GP predictions based on the inducing outputs. Note that respecting the conditional dependence between $x_{0:T}$ and F_M is important for accurate inference but prevents us from analytically marginalizing out the inducing outputs (Ialongo et al., 2019) as is done in standard sparse GP regression. We further chose as variational distribution over the inducing outputs $q(f_M^{(l)}) = \mathcal{N}\left(f_M^{(l)} \mid m_M^{(l)}, \mathrm{S}_M^{(l)}\right)$ with free parameters $m_M^{(l)} \in \mathbb{R}^M, \mathrm{S}_M^{(l)} \in \mathbb{R}^{M \times M}$, and over the initial latent states $q(x_0^{(l)}) = \mathcal{N}\left(x_0^{(l)} \mid m_0^{(l)}, \mathrm{S}_0^{(l)}\right)$, with free parameters $m_0^{(l)} \in \mathbb{R}^{D_l}, \mathrm{S}_0^{(l)} \in \mathbb{R}^{D_l \times D_l}$. More flexible recognition models can easily be incorporated (Doerr et al., 2018).

Variational Inference We want to find the optimal values for the variational parameters that minimize the KL divergence between the approximate posterior $q(\cdot)$ and the true posterior $p(\cdot|y_T)$. Analogously, we can maximize the lower bound $\mathcal{L}$ to the log marginal likelihood (Blei et al., 2017):

$$\mathcal{L} = \mathbb{E}_{q(x_{0:T}, f_M)}\left[\log \frac{p(x_{0:T}, f_M, y_{1:T})}{q(x_{0:T}, f_M)}\right] \tag{12}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q(x_t)}\left[\log p\left(y_t \mid x_t\right)\right] - \mathrm{KL}\left(q\left(x_0\right) \middle\| p\left(x_0\right)\right) - \mathrm{KL}\left(q\left(f_M\right) \middle\| p\left(f_M\right)\right), \tag{13}$$

where Eq. (13) results from plugging Eqs. (10) and (11) into Eq. (12). Here $q(x_0) = \prod_{l=1}^{L} q(x_0^{(l)})$ and analogously $p(x_0)$, $q(f_M)$ and $p(f_M)$. The marginal $q(x_t) = \prod_{l=1}^{L} q(x_t^{(l)})$ decomposes between the components with $q(x_t^{(l)}) = \int q(x_t^{(l)}|f_M^{(l)}) q(f_M^{(l)}) df_M^{(l)}$ and $q(x_t^{(l)}|f_M^{(l)}) = \int q(x_0^{(l)}) \prod_{t'=0}^{t-1} p(x_{t'+1}^{(l)}|x_{t'}^{(l)}, f_M^{(l)}) \prod_{t'=0}^{t-1} dx_{t'}^{(l)}$. As a remark, the variational distribution $q(x_t^{(l)})$ has no closed-form solution and can be combined with different Monte Carlo sampling strategies as presented in Longi et al. (2021). In our experiments, we adopt the scheme from Ialongo et al. (2019) since it leads to unbiased samples and scales linearly with $O(t)$. Finally, both KL-divergences can be computed in closed-form since all involved distributions are Gaussians.

Backfitting Algorithm Since the variational posterior [Eq. (11)] decomposes between the components, we can apply an iterative learning algorithm for parameter optimization. The backfitting algorithm (Breiman and Friedman, 1985) cycles through all L components to find the optimal set of parameters $\Theta = \{\theta^{(l)}\}_{l=1}^{L}$ where $\theta^{(l)}$ consists of the variational parameters $\{m_0^{(l)}, S_0^{(l)}, m_M^{(l)}, S_M^{(l)}\}$ and the hyperparameters belonging to the l-th component (e.g. GP kernel parameters, inducing inputs). In each step, we perform an inner optimization to update the parameters $\theta^{(l)}$ of the l-th component, while keeping all other parameters $\Theta \setminus \theta^{(l)}$ fixed. The emission output matrix C is not assigned to any component and updated in every optimization step. While the benefits of a sequential learning scheme might not be clear yet, we will exploit its assumptions in the subsequent section to learn the parameters $\theta^{(l)}$ of each component with a different resolution in order to capture effects on multiple time scales.

Mini-Batching Since the lower bound $\mathcal{L}$ decomposes between the time points, we can obtain an unbiased estimate using only a subset of the sequence (Bottou, 2010), $\sum_{t=1}^{T} \mathbb{E}_{q(x_t)}\left[\log p\left(y_t \mid x_t\right)\right] \approx \frac{T}{B} \sum_{t=t_0}^{B+t_0} \mathbb{E}_{q(x_t)}\left[\log p\left(y_t \mid x_t\right)\right]$ where B is the batch size and t_0 denotes the first time index in the batch. To sample efficiently from the marginal $q\left(x_t\right)$, we make one rather common approximation (Aicher et al., 2019): We break the temporal dependency between x_t, and its predecessors $x_0, \ldots, x_{t_0-B_0}$, where B_0 is the buffer size, by sampling $x_{t_0-B_0}$ directly from the recognition model, $q(x_0)$. Together with the reparameterization trick (Kingma and Welling, 2013), we can exploit this subsampling scheme for computing cheap gradients during parameter optimization. However, breaking the temporal dependency also leads to biased gradients: effects that evolve slower than the size of the mini-batch can no longer be inferred.

In principle, one could resolve this issue by downsampling the data in a preprocessing step. However, this comes at the expense of fast varying dynamics that can then no longer be modeled (see Figure 1). We compare to this approach in our experiments.

3.3. Training over Multiple Resolutions

Prior work on GPSSMs takes only the dynamics of a single resolution into account which is not sufficient if effects on multiple time scales are present. To circumvent this shortcoming, we proceed by interpreting the GP transition model through the lens of SDEs.

Relationship to SDEs Consider multi-component state-space models in which the transition model of the l-th component is given by

$$p^{\Delta}(x_{j+1}^{(l)}|x_j^{(l)}, f_M^{(l)}) = \mathcal{N}(x_{j+1}^{(l)}|x_j^{(l)} + R\Delta_t\mu^{\Delta}(x_j^{(l)}), \quad (R\Delta_t)^2\Sigma^{\Delta}(x_j^{(l)}) + R\Delta_t\mathrm{Q}^{\Delta}), \tag{14}$$

where $\mu^{\Delta}(x_j^{(l)})$ and $\Sigma^{\Delta}(x_j^{(l)})$ are the equivalents of the GP prediction in Eq. (1), and we again marginalized the local latent variables $f_j^{(l)}$ out of the discretized SDE [Eqs. (6) and (7)] using Gaussian calculus. After restricting $R \geq 1$ to be integer, we define all remaining terms of the model and the structured variational family analogously as in Eqs. (10) and (11), leading to the lower bound

$$\mathcal{L}_{\Delta} = \sum_{j=1}^{J} \mathbb{E}_{q^{\Delta}(x_j)}\left[\log p^{\Delta}(y_j \mid x_j)\right] - \mathrm{KL}\left(q^{\Delta}(x_0)\middle\|p^{\Delta}(x_0)\right) - \mathrm{KL}\left(q^{\Delta}\left(f_M\right)\middle\|p^{\Delta}\left(f_M\right)\right), \tag{15}$$

where $J = T/R$. We next present the equivalence between the GP and discretized SDE formulations for $R = 1$, i.e. for equal time steps.

Theorem 1 *For $R = 1$, there exists a setting of the model and variational parameters of the SDE formulation in terms of those of the GP formulation such that $\mathcal{L} = \mathcal{L}_{\Delta}$.*

We give the exact parameterization and a constructive proof in an extended version (Longi et al., 2021). We first provide the analytical formulae for the marginalization over the inducing outputs f_M in the SDE and in the GPSSM formulation. After showing that these formulae are consistent, we show that this consistency is passed on to the evidence lower bound.

Our findings allow us to reinterpret the GP transition model [Eq. (4)] as a discretized SDE with $R = 1$. Choosing a resolution $R > 1$, we can approximate the GPSSM lower bound [Eq. (13)] using the SDE formulation [Eq. (15)]. The novelty in contrast to other works using the connection between GPSSMs and discretized SDEs (e.g. Ruttor et al. (2013), Zhao et al. (2020)) is that we draw an additional connection to canonical GPSSMs and exploit it for training the latter with multiple resolutions by applying different approximation levels R. In the following, we take this to our advantage in order to come up with an efficient algorithm to learn effects on multiple time scales.

Multi-Resolution Learning Our algorithm decomposes the dynamics into L components corresponding to different time scales. The components are fit iteratively by using the backfitting algorithm whereby each component is inferred with a different resolution. For training the components of lower resolutions, we dilate the minibatch scheme by taking only every R-th observation into account in order to load larger histories into a mini-batch of fixed size B. However, naively computing the marginal $q(x_{t_0+BR}^{(l)})$ would be too expensive since it requires BR sampling steps. We can overcome this issue by interpreting the component under the SDE perspective with resolution level R using the lower bound [Eq. (15)] which allows us to draw instead B approximate samples from $q^{\Delta}(\cdot)$ [Eq. (14)]. Hence, we can approximate the lower bound at different resolution levels using $\sum_{t=1}^{T} \mathbb{E}_{q(x_t)}\left[\log p\left(y_t \mid x_t\right)\right] \approx \frac{T}{B}\sum_{j=j_0}^{B+j_0} \mathbb{E}_{q^{\Delta}(x_j)}\left[\log p^{\Delta}\left(y_j \mid x_j\right)\right]$. This approximation keeps the

runtime fixed over different resolutions, while the approximation level of the marginal is adjusted to the resolution level of the component under consideration. Fast transitions are captured by high-resolution components with tight discretization levels ($R = 1$), while slow transitions are captured by low-resolution components with long histories ($R > 1$).

Since our variational family assumes that the latents are independent between components [Eq. (11)], we can compute the simulated latents of all but the l-th component, $x_{0:T}^{(\neq l)}$, outside of the inner optimization scheme of the backfitting algorithm. The latter leads not only to a reduction in runtime, but also enables the use of different resolution levels across components in order to ensure that the discretization level is sufficiently tight for fast dynamics and the history length is sufficiently long for slow dynamics. We detail out the algorithm and provide its runtime analysis in an extended version of this paper (Longi et al., 2021).

Limitations We build on the variational family of Doerr et al. (2018), that uses the prior $p(x_t^{(l)}|x_{t-1}^{(l)}, f_M)$ as approximate smoothing distribution $q(x_t^{(l)}|\cdot)$. While extensions to more complex variational posteriors exist, they do not allow for mini-batching (Ialongo et al., 2019) or make strong independence assumptions on $q(f_M^{(l)}, x_{1:T}^{(l)})$ (e.g. Eleftheriadis et al., 2017). The methodological novelty of our work is to a large extent agnostic to the choice of $q(x_t^{(l)}|\cdot)$ and we expect that improvements on the inference scheme for general GPSSMs can be easily combined with our work.

4. Experiments

We validate the presented algorithm on semi-synthetic data and on an emission modeling task, confirming that using multiple resolutions compares favorably to state-of-the-art methods that operate on a single resolution only. We compare our novel multi-resolution GPSSM (MR-GPSSM) against the standard GPSSM applying a similar inference scheme (Doerr et al., 2018). This approach uses a non-structured latent space which does not allow for learning on multiple resolutions. To tease apart the effects of multiple components and multiple resolutions, we additionally introduced the multi-component GPSSM (MC-GPSSM). The latter has the same architecture and employs the same optimization algorithm as MR-GPSSM, but applies a single resolution over all components. We refrained from benchmarking against other non state-space GP models since this has already been done extensively in Doerr et al. (2018), demonstrating the benefits of their method that we compare against. We measure the performance via the root mean squared error (RMSE) and report the mean and the standard error over five runs. Many more experimental details and results can be found in the extended version of this paper (Longi et al., 2021). Code is available at https://version.helsinki.fi/MUPI/mr-gpssm.

4.1. Semi-Synthetic Data

First, we benchmarked our method on 4 semi-synthetic datasets with varying properties: fast dynamics (F), mixed dynamics (M1, M2), and slow dynamics (S). Dataset M1 and M2 exhibit both fast and slow dynamics, and are challenging for previous methods. Each dataset consists of $T = 37,961$ time points, from which we used the first half for training and the second half for testing.

For MR-GPSSM, we applied $L = 2$ components with $D_x = 2$ latent dimensions each, and learned one component with $R^{(f)} = 1$ for fast dynamics and one with $R^{(s)} = 30$ for slow dynamics. We trained each component for 600 iterations that were split evenly into 12 backfitting cycles. We compared our model to MC-GPSSM using exactly the same settings. For standard GPSSM, we set

Table 1: **Results on Semi-Synthetic Data.** Predictive performance of GPSSM variants on four semi-synthetic datasets with varying dynamics: slow (S), mixed (M1, M2) and fast (F). The best performing method, and all methods whose mean statistic overlap within the standard error, are marked in bold.

		GPSSM		**MC-GPSSM**		**MR-GPSSM** (ours)
		$R = 1$	$R = 30$	$R = [1, 1]$	$R = [30, 30]$	$R = [30, 1]$
RMSE	**F**	**0.05 (0.00)**	0.14 (0.00)	**0.06 (0.01)**	0.16 (0.01)	0.07 (0.01)
	M1	0.16 (0.02)	0.14 (0.00)	0.15 (0.01)	0.15 (0.00)	**0.08 (0.01)**
	M2	0.14 (0.00)	0.29 (0.11)	0.14 (0.00)	0.20 (0.01)	**0.09 (0.01)**
	S	0.33 (0.08)	**0.16 (0.01)**	0.29 (0.02)	0.20 (0.03)	**0.17 (0.02)**

the number of latent states to $D_x = 4$ and trained for 600 iterations such that the model complexity and the number of parameter updates is comparable. We varied the resolution for both comparison partners in $R \in \{1, 30\}$. The results are shown in Table 1. We observe that MC-GPSSM and GPSSM perform well if the resolution is chosen appropriately: Fast dynamics (dataset F) can only be accurately predicted using a small resolution ($R = 1$), whereas slow dynamics (dataset S) require a large resolution ($R = 30$). Moreover, choosing the wrong resolution leads not only to a decrease in performance, but also to convergence problems which lead to the removal of one run of GPSSM ($R = 1$) on dataset S. Our proposed model, MR-GPSSM, achieves comparable results on both tasks. On datasets with mixed dynamics (M1, M2), MR-GPSSM significantly improves over the single resolution models, since it is the only method that captures effects on multiple timescales.

Next, we investigated if increasing the mini-batch size can provide an alternative solution for capturing slow dynamics. Instead of learning the dynamics with resolution $R = 30$ and minibatch size $B = 50$, as done previously, we increased the mini-batch size to $B = 1500$ and applied the standard resolution $R = 1$. The results in Longi et al. (2021) confirm on dataset S that the latter strategy does not yield competitive results even if we allow for prolonged training time.

4.2. Engine Modeling Task

This dataset consists of 22 independent measurements containing the raw emissions of an engine. Each measurement is recorded with 10Hz and between 21 and 63 minutes long, resulting in over 500,000 data points. The system is described by 4 inputs and the following 4 outputs: particle numbers (PN), hydrocarbon concentration (HC), nitrogen oxide concentration (NOx) and engine temperature (Temp). In the following, we split the data into 16 train and 6 test measurements. For each output, the experiment is carried out 5 times using stratified cross-validation. Our early results indicated that the optimization can be sensitive to the initial conditions when the mini-batches are not chosen small enough or the used resolution is not adquate. In order to avoid local optima, we repeated each training 3 times using random restarts, and selected the model with the best training objective for predicting on the test set.

First, we studied if the optimal resolution differs between outputs by performing a grid search over $R \in \{1, 5, 10, 20, 30, 40, 50, 60, 70\}$ using standard GPSSM. We set the number of latent dimensions to $D_x = 6$ and use $3,000$ training iterations. The results are shown in the extended manuscript (Longi et al., 2021) and, for the found optimal resolutions $R = \{1, 5, 30\}$ in Figure 3. Next, we trained MR/MC-GPSSM using a comparable configuration ($L = 3$; $D_x = 2$; $3,000$

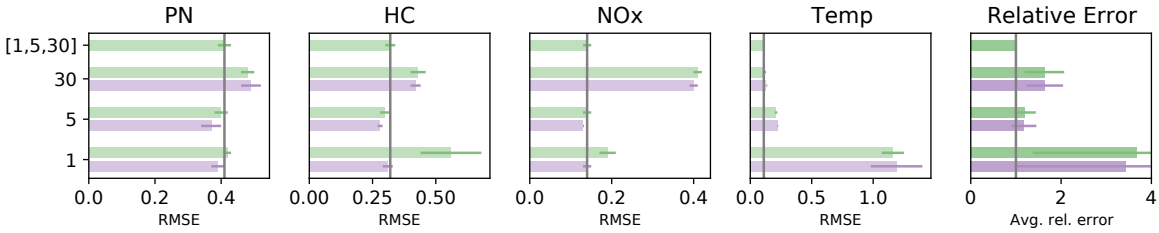

Figure 3: **Predictive Performance on Engine Modeling Task.** RMSE on the four outputs PN, HC, NOx, Temp and relative error with respect to our method, averaged over all outputs. From top to bottom: We compare our method, MR-GPSSM (green, indicated with a gray line), to MC-GPSSM (green) and GPSSM (purple) sorted according to decreasing resolution. Our method is the only method that performs consistently well over all outputs.

Table 2: **Sensitivity analysis.** Predictive performance of MR-GPSSM on emission datasets when varying the resolution set. The first column $R = [30, 5, 1]$ corresponds to the default setting. We can observe that the performance is consistent when varying the size of the resolution set. The outputs HC and NOx require at least one component with small resolution, while the output Temp requires at least one component with large resolution.

R	$[30, 5, 1]$	$[7, 5, 1]$	$[40, 20, 10]$	$[40, 20]$	$[30, 15, 7, 5, 1]$	$[40, 20, 10, 5, 1]$
PN	0.41 (0.02)	0.40 (0.03)	0.41 (0.02)	0.45 (0.02)	0.40 (0.02)	0.39 (0.02)
HC	0.32 (0.02)	0.29 (0.01)	0.38 (0.03)	0.40 (0.02)	0.31 (0.02)	0.32 (0.01)
NOx	0.14 (0.01)	0.15 (0.01)	0.19 (0.01)	0.33 (0.01)	0.15 (0.01)	0.17 (0.02)
Temp	0.11 (0.01)	0.22 (0.03)	0.10 (0.00)	0.11 (0.00)	0.10 (0.00)	0.10 (0.00)

iterations per component). We set the resolutions of MR-GPSSM to $R = [1, 5, 30]$ such that the best resolution for each output is included, and trained MC-GPSSM on each resolution independently. The results are shown in Figure 3. Our method, MR-GPSSM, shows competitive performance across all outputs, while (MC-)GPSSM works only well if the resolution is set adequately. In addition, MR-GPSSM requires less fine-tuning, and also performs well if the resolution set is varied as shown in Table 2.

5. Conclusion

We have presented a novel Gaussian Process state-space model architecture that allows to traverse time with multiple resolutions. It is composed of multiple components that evolve independently over time. By interpreting the transition functions as discretized stochastic differential equations, we can learn each component with a different resolution to model effects on different time scales.

The benefits of our approach are demonstrated on semi-synthetic data and on a challenging engine modeling task. However, our methodological contribution is general and can also be applied to use cases from different domains ranging from neuroscience (Prince et al., 2021), medicine (Lipton et al., 2016) to human motion prediction (Martinez et al., 2017).

References

Christopher Aicher, Srshti Putcha, Christopher Nemeth, Paul Fearnhead, and Emily B Fox. Stochastic gradient mcmc for nonlinear state space models. *arX iv preprint arXiv:1901.10568*, 2019.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 2017.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings in Computational Statistics*, 2010.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 1985.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. *International Conference on Machine Learning*, 2011.

Andreas Doerr, Christian Daniel, Martin Schiegg, Duy Nguyen-Tuong, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Probabilistic recurrent state-space models. *International Conference on Machine Learning*, 2018.

Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of gaussian process state space models. *Advances in Neural Information Processing Systems*, 2017.

Roger Frigola. *Bayesian time series learning with Gaussian processes*. PhD thesis, University of Cambridge, 2015.

James B Heaton, Nick G Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 2017.

Pashupati Hegde, Markus Heinonen, Harri Lähdesmäki, and Samuel Kaski. Deep learning with differential gaussian process flows. *International Conference on Artificial Intelligence and Statistic*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. LSTM can solve hard long time lag problems. *Advances in Neural Information Processing Systems*, 1997.

Alessandro Davide Ialongo, Mark Van Der Wilk, James Hensman, and Carl Edward Rasmussen. Overcoming mean-field approximations in recurrent gaussian process models. *International Conference on Machine Learning*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *International Conference on Learning Representations*, 2016.

Krista Longi, Jakob Lindinger, Olaf Duennbier, Melih Kandemir, Arto Klami, and Barbara Rakitsch. Traversing time with multi-resolution gaussian process state-space models. *arXiv preprint arXiv:2112.03230*, 2021.

Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*, 2013.

Luke Yuri Prince, Shahab Bakhtiari, Colleen J Gillon, and Blake A Richards. Parallel inference of hierarchical latent dynamics in two-photon calcium imaging of neuronal populations. *arXiv preprint arXiv:1803.01271*, 2021.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT press Cambridge, 2006.

Andreas Ruttor, Philipp Batz, and Manfred Opper. Approximate gaussian process inference for the drift function in stochastic differential equations. *Advances in Neural Information Processing Systems*, 2013.

Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*. Cambridge University Press, 2019.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 2005.

Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. *Advances in Neural Information Processing Systems*, 2005.

Ronald J Williams and David Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. *Backpropagation: Theory, architectures, and applications*, 1995.

Cagatay Yildiz, Markus Heinonen, Jukka Intosalmi, Henrik Mannerstrom, and Harri Lahdesmaki. Learning stochastic differential equations with gaussian processes without gradient matching. *International Workshop on Machine Learning for Signal Processing*, 2018.

Changmin Yu, Marko Seslija, George Brownbridge, Sebastian Mosbach, Markus Kraft, Mohammad Parsi, Mark Davis, Vivian Page, and Amit Bhave. Deep kernel learning approach to engine emissions modeling. *Data-Centric Engineering*, 2020.

Zheng Zhao, Filip Tronarp, Roland Hostettler, and Simo Särkkä. State-space gaussian process for drift estimation in stochastic differential equations. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

Christoph Zimmer and Reza Yaesoubi. Influenza forecasting framework based on gaussian processes. *International Conference of Machine Learning*, 2020.