

---

# Aligned Multi-Task Gaussian Process

---

Olga Mikheeva<sup>1</sup>, Ieva Kazlauskaitė<sup>2</sup>, Adam Hartshorne<sup>3</sup>, Hedvig Kjellström<sup>1</sup>,  
Carl Henrik Ek<sup>2</sup>, Neill D. F. Campbell<sup>3</sup>

<sup>1</sup> KTH Royal Institute of Technology

<sup>2</sup> University of Cambridge

<sup>3</sup> University of Bath

## Abstract

Multi-task learning requires accurate identification of the correlations between tasks. In real-world time-series, tasks are rarely perfectly temporally aligned; traditional multi-task models do not account for this and subsequent errors in correlation estimation will result in poor predictive performance and uncertainty quantification. We introduce a method that automatically accounts for temporal misalignment in a unified generative model that improves predictive performance. Our method uses Gaussian processes (GPs) to model the correlations both within and between the tasks. Building on the previous work by Kazlauskaitė et al. (2019), we include a separate monotonic warp of the input data to model temporal misalignment. In contrast to previous work, we formulate a lower bound that accounts for uncertainty in both the estimates of the warping process and the underlying functions. Also, our new take on a monotonic stochastic process, with efficient path-wise sampling for the warp functions, allows us to perform full Bayesian inference in the model rather than MAP estimates. Missing data experiments, on synthetic and real time-series, demonstrate the advantages of accounting for misalignments (vs standard unaligned method) as well as modelling the uncertainty in the warping process (vs baseline MAP alignment approach).

## 1 Introduction

Multivariate datasets gathered across a range of tasks are increasingly prevalent. In contrast to the estab-

lished regression regime, where we aim to learn correlations across a time series within a single data source, we now wish to consider the relationships between different sources of data. This is the canonical multi-task learning scenario where we seek to model both the correlations within individual datasets as well as the correlations between datasets. If we perform this well, we are able to provide high-quality predictions, with appropriate uncertainty quantification, under missing data scenarios; we can use correlations between time-series to fill in the gaps in data for specific instances. Success necessitates an accurate decomposition of correlations between these two factors and is inherently ill-posed.

To make progress we must find a principled regularisation that trades-off between the two generating components. Current approaches suffer a limiting assumption that all sources of data have perfect temporal alignment. Importantly, our terminology refers to the fundamental alignment between the generative process, not to the precision of a sampling rate. For example, two sources of data can share a common ancestral generating process but subsequent activities will introduce delays and phase shifts that result in temporal misalignment irrespective of some measurement clock. Failure to account jointly for these effects must lead to incorrect estimates of task correlations; this is particularly noticeable when we seek to account for uncertainty in our predictions.

This problem is also called domain shift; the observed covariates are transformed from some idealised covariates via a distinct (unknown) per-task transformation (Quionero-Candela et al., 2009). The idealised covariates are typically unobserved, making such varying shifts challenging to identify. The problem is further complicated by the i.i.d. observation noise.<sup>1</sup> Typical examples of knowledge transfer between time-series data in a regression setting include: observing data

---

Proceedings of the 25<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

---

<sup>1</sup>Knowledge transfer between tasks is only possible when observations are assumed to include observation noise or correspond to different inputs (Wackernagel, 2003; Bonilla et al., 2008; Álvarez and Lawrence, 2011).

from multiple tasks and sharing the knowledge between tasks (Bonilla et al., 2008); observing multiple trials of the same experiment and inferring missing data in some of the trials using the information from the others (Álvarez et al., 2012); multi-fidelity learning using cheap measurements as a proxy for expensive ones (Liu et al., 2018; Wang et al., 2020); and clinical bench-marking where the emphasis is on the interpretability of the parameters of the covariates (Dürichen et al., 2014).

This paper addresses the temporal misalignment problem in a multi-task setting for time-series data. We build our model on Gaussian processes (GPs) to encode the prior knowledge of the inter-task and intra-task structure of the observations. We use a latent variable construction to infer the inter-task correlations. Similarly to the GP-LVA model of Kazlauskaitė et al. (2019), we introduce a separate warping function for each task to address dataset shift. We systematically remove the mismatch between tasks and allow the multi-task model to accurately transfer information between them.

Fig. 1 illustrates a typical problem of dataset shift, *i.e.* the inputs to each sequence have been warped by unknown functions, with missing data. Our aligned multi-task Gaussian process (AMTGP) model shares the inter-task information to compensate for both the unknown transformation of the inputs and the missing data. The inferred uncertainty in the unknown warps, Fig. 1(c), is observed to correlate with the regions of missing data and prevents overconfidence. In our experiments, we test the model on synthetic and real time-series. We demonstrate that information sharing between tasks is improved by aligning the inputs and quantify the performance of the standard and the aligned MTGP models on missing data problems. We show that our uncertainty estimates are superior to the previous GP-LVA approach.

In summary, the contributions of this work are: (1) a novel probabilistic approach for information transfer between tasks corrupted by temporal misalignment; (2) an efficient inference scheme based on sparse stochastic variational inference; (3) a reformulation of monotonic GP flow (Ustyuzhaninov et al., 2020) with efficient training; and (4) a model that is a generalisation of the GP-LVA model with a rigorous probabilistic formulation.

## 2 Related Work

GPs are a standard Bayesian tool for time-series problems and have been used in Multi-Task (MT) settings such as dependent GPs (Boyle and Frean, 2004), Multi-Output GPs (MOGPs) (Bilonis et al.,

2013; Álvarez et al., 2012) and Multi-Task GPs (MT-GPs) (Bonilla et al., 2008). Historically, the topic has been termed the linear model of coregionalization (Journal and Huijbregts, 1978), kernel methods for vector-valued functions (Evgeniou and Pontil, 2004) and matrix-variate Gaussian distributions (Dutilleul, 1999); for a review please see, *e.g.* (Álvarez et al., 2011).

**MTGPs** Bonilla et al. (2008) place a GP prior over each sequence (task) and include a free-form covariance matrix (constrained to be positive semi-definite). To reduce complexity, inter-task correlations can be modelled using probabilistic principal component analysis (pPCA); Stegle et al. (2011) extend this to a GP-LVM for the covariance. Álvarez and Lawrence (2011) use convolution processes to impose correlations that can be applied in cases where some of the sequences are blurred versions of the others. More recently, Boustati et al. (2019) used compositional (deep) GPs for MT learning through non-linear mixing of latent processes (shared and individual). Zhe et al. (2019) propose a MOGP model with latent GPs as covariates and focus on inference efficiency exploiting the grid placement of the observations and the (deep) Kronecker factorisation. Hamelijncx et al. (2019) propose an application of the MT framework to multi-resolution spatio-temporal problems.

**Alignment** While many heuristic methods for the alignment of data have been developed (*e.g.* dynamic time warping), the work on alignment of data in probabilistic multi-task learning has been limited to simple shift or scale (Dürichen et al., 2014). The multi-task models from statistical literature largely come from geostatistics, *i.e.* spatio-temporal modelling (Sahu and Mardia, 2005). These models are usually application specific and in most cases use linear transformations for alignments (*e.g.* (Forlani et al., 2020)). Another line of work is structured covariance estimation (Barnard et al., 2000), (Spezia, 2019), although these methods do not explicitly model misalignments.

**Deep GPs & Alignment** While some of the proposed approaches consider deep GPs (DGPs) as models for sequences (Boustati et al., 2019; Hamelijncx et al., 2019), their motivation is applications where the data are known to be generated by functional composition. Explicit temporal mismatch between sequences is not considered (no monotonic constraint on latent layers). Importantly, the existing works that do model temporal alignment, (Kaiser et al., 2018; Duncker and Sahani, 2018), assume the groups of tasks to align are known a-priori. This significantly simplifies the alignment problem and this knowledge is not present in

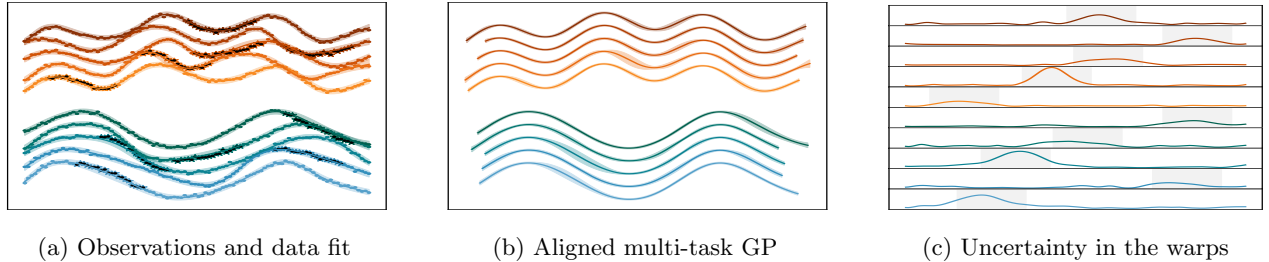


Figure 1: Multi-task model for 10 time-series with missing data (shown in black). (a) Observed data that comes from two different underlying sequences; for clarity, the sequences are coloured and plotted with a vertical offset. (b) Fitted aligned multi-task model. The model correctly uncovers and describes the two types of sequences despite missing data and dataset shift. (c) The predictive standard deviation (uncertainty) of the estimated warps is shown to correlate well with the missing data regions (shown in grey) for each task.

the general MT learning formulation. The motivation for our work is closer to the GP alignment models (Kazlauskaitė et al., 2018, 2019). Contrary to these models, we propose a joint probabilistic approach that is motivated by MT applications rather than a regularised GP regression model that is aimed primarily at an alignment goal.

**Sparse Variational GPs** One of the weaknesses of the traditional GP formulation is the poor computational scaling with respect to the number of observations; this is especially apparent in the multi-task case where the computational cost scales as  $\mathcal{O}(J^3 N^3)$  for  $J$  tasks, each with  $N$  observations. Therefore, efficiency issues of such multi-task models have been considered in most papers on the subject, *e.g.* (Álvarez et al., 2012; Hamelijncx et al., 2019; Zhe et al., 2019).

Bringing the ideas from Kazlauskaitė et al. (2018) to a multi-task scenario, we propose a method that is able to model flexibly misalignments in GP multi-task learning. In this work, we follow the sparse GP approach of Titsias (2009) and the subsequent stochastic extension of the variational inference framework (Hensman et al., 2013).

### 3 Background

**Gaussian Processes** We make use of GPs for the Bayesian modelling of time-series data as they offer a convenient way of defining priors over functions (Williams and Rasmussen, 2006). We denote a GP functional prior, fully specified by a mean function  $m(x)$  (typically assumed to be zero) and a covariance function  $k(x, x')$ , as  $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$ . Thus given a finite set of inputs  $x_1, \dots, x_N$ , we may draw samples  $f(x_1), \dots, f(x_N)$  from the GP prior:  $f(x_1), \dots, f(x_N) \sim \mathcal{N}(0, \mathbf{K})$  where  $K_{ij} := k(x_i, x_j)$ . The model of the data is  $y_i = f(x_i) + \epsilon_i$  where  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$  is Gaussian noise. Learning in exact GP models typically consists of inferring the hyper-

parameters of a specified covariance function. This can only be performed in closed form under Gaussian likelihoods and at high computational expense  $\mathcal{O}(N^3)$  (due to the inversion of the covariance matrix); approximate inference methods provide more efficient inference and relax the likelihood restrictions.

**Multi-task GPs** In multi-task GPs (MTGPs), we assume that observations of some latent functions  $\mathbf{F} \in \mathbb{R}^{N \times J}$  comprise  $J$  sequences (corresponding to  $J$  tasks), each of which we model using a GP and, furthermore, there exists some unknown correlation structure among the tasks. In one of the most widely used models, intrinsic coregionalization model (ICM, Goovaerts et al. (1997)), the joint model is then defined as  $\text{vec}(\mathbf{F}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{K}_\psi \otimes \mathbf{K}_\theta)$  with some mean  $\mathbf{M}$  and a covariance structure that adopts a Kronecker product form where the  $J \times J$  covariance matrix  $\mathbf{K}_\psi$  captures the correlations among the  $J$  tasks, while the  $N \times N$  matrix  $\mathbf{K}_\theta$  models the correlations between the  $N$  observations in each of the  $J$  sequences.

While we base our work on ICM and its latent variable extension, the idea of temporal alignment in multi-task learning is general and can be applied to other MTGP models. For a review of other MTGP models see Liu et al. (2018).

**GP Latent Variable Alignment Model** Here we give a brief introduction to the work of Kazlauskaitė et al. (2019), which inspired this work, and highlight the important differences. GP Latent Variable Alignment (GP-LVA) (Kazlauskaitė et al., 2018, 2019) is designed with the primary goal of sequence alignment. The method models aligned functions as pseudo-observations that should have high likelihoods under two separate parts of the model simultaneously. One part models temporal consistency using GPs, the other models inter-sequence relationship using a GP-LVM. While this trick works empirically, it has a number of downsides: (1) predictive posterior is condi-

tioned on both data and pseudo-observations, which leads to underestimated uncertainty, (2) it is unclear how to generate samples from the model since the dependencies between two parts of the model are induced via the pseudo-observations of aligned functions, while marginalizing out these aligned functions leads to the two parts of the model becoming independent, and (3) to reconcile the two parts of the model the authors add heuristic noise terms which reduces the model interpretability. In contrast, our model is a fully generative probabilistic model, cast in terms of a standard GP framework. We formulate a proper evidence lower bound, used for inference and hyperparameter learning. The aligned functions are treated in the Bayesian way, with variational posterior distributions and the latent warpings are explicitly modelled and marginalised out (rather than taking a MAP estimate as in GP-LVA). In summary, the model of Kazlauskaitė et al. (2019) could also be seen as a partial approximation to our model while we preserve the full model and use approximate inference. Moreover, due to its use of pseudo-observations, GP-LVA can not handle missing data in a principled way, suffering from underestimated uncertainty, particularly in the areas of missing data.

**Monotonic GPs** To model the aligned functions we must account explicitly for the unknown misalignment subject to the constraint that it must be monotone. There have been a number of proposals for approximations to define monotonic GPs: for example, truncated or finite-dimensional approximations (Maatouk, 2017; Lopez-Lopera et al., 2019); incorporating virtual derivative information (Riihimäki and Vehtari, 2010); projections onto spaces of monotone functions (Lin and Dunson, 2014); or through non-linear transformations (Andersen et al., 2018). Instead, we define a guaranteed monotonic stochastic process through a differential flow that provides smooth solutions that are guaranteed monotonic across the entire domain without distorted uncertainty estimation (Ustyuzhaninov et al., 2020). We extend this work with a modified model, better considered as an Ordinary Differential Equation (ODE) with an uncertain drift function, rather than the Stochastic Differential Equation (SDE) of Ustyuzhaninov et al. (2020), and provide a new approach for efficient inference.

## 4 Model

Consider a data vector  $\mathbf{y}_j = \{y_{jn}\}_{n=1}^N$ , where  $y_{jn}$  is a noisy observation of the function  $f_j(\mathbf{x}_{jn})$  and a corresponding input vector  $\mathbf{x}_j = \{\mathbf{x}_{jn}\}_{n=1}^N$  for each of  $J$  tasks with a length of  $N$  observations. For clarity of notation, we will consider tasks to be of the same

length; in the case of different lengths, the `vec` operator should be replaced with concatenation. Let  $\mathbf{f}_j$  denote the values of the function  $f_j(\cdot)$  at inputs  $\mathbf{x}_j$ . We denote all input data as  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_J]$  and the observed data matrix as  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_J]$ . Stacked vectors of the observed data and inputs is then denoted as  $\mathbf{y} = \text{vec}(\mathbf{Y})$  and  $\mathbf{x} = \text{vec}(\mathbf{X})$  respectively. Similarly,  $\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_J]$  and  $\mathbf{f} = \text{vec}(\mathbf{F})$ .

### 4.1 Multi-Task Gaussian Processes (MTGPs)

First, we introduce a latent variable version of the standard MTGP formulation. Typically, in ICM the correlations between tasks are modelled with a free-form covariance (in the absence of task-specific features). Similarly to Stegle et al. (2011) and Dai et al. (2017), we choose a more flexible approach and use latent variables to model the inter-task dependencies. Each task is assumed to have a corresponding latent variable  $\mathbf{z}_j \in \mathbb{R}^Q$ . We put a spherical Gaussian prior on the latent variables  $\mathbf{z}_j \sim \mathcal{N}(\mathbf{z}_j | \mathbf{0}, \mathbf{I})$ . The functions are assumed to be a joint sample from a GP with a separable kernel over the latent and input spaces taking the form  $\mathbf{f} | \mathbf{z}, \mathbf{X} \sim \mathcal{GP}(\mathbf{f} | \mathbf{0}, \mathbf{K})$ , where  $K_{jn,j'n'} = k_\psi(\mathbf{z}_j, \mathbf{z}_{j'}) k_\theta(\mathbf{x}_n, \mathbf{x}_{n'})$  is the covariance between the  $n$ -th input of  $\mathbf{f}_j(\cdot)$  and the  $n'$ -th input of  $\mathbf{f}_{j'}(\cdot)$ . The kernel  $k_\psi(\cdot, \cdot)$  acting on latent variables determines similarities across the tasks, and kernel  $k_\theta(\cdot, \cdot)$  affects the properties of each of the tasks separately. This approach allows for the explicit incorporation of priors on the inter-task dependencies.

### 4.2 Aligned Multi-Task GPs (AMTGPs)

In a standard MTGP model, the tasks are assumed to be *aligned* across inputs. However, especially in realistic scenarios where the input is time, these tasks might be misaligned due to various unmodelled factors. To overcome this, we account for temporal misalignments between tasks by warping the inputs with latent monotonic functions; this reflects the assumption that misalignment manifests as delays and phase shifts but not as non-causal permutations of time. The values of  $\mathbf{f}_j$  are modelled using inferred aligned input values  $\mathbf{g}_j$ .

**Monotonic Warps** Here we consider warps to be independent between tasks (*e.g.* sampling errors or phase noise) that we model as smooth monotonic functions. For each task  $j$ , the alignment is modelled with a monotonic function  $g_j(x_{jn})$  and the corresponding values of the function for all inputs  $\mathbf{x}_j$  are denoted by  $\mathbf{g}_j$ . As discussed in Sec. 3, there are a number of different approaches to constrain a GP to be monotonic. We build upon the monotonic GP flow solution proposed by Ustyuzhaninov et al. (2020). There, a stochastic process is defined as a fixed time, initial value solu-

tion to a Stochastic Differential Equation (SDE). Subject to constraints on how the SDE is defined (using a GP field) and the inference procedure, every sampled solution is guaranteed not to permute the inputs and, therefore, remain monotone. In contrast to the SDE formulation, we pose a monotonic process as the solution to an Ordinary Differential Equation (ODE)  $du = w(u) d\tau$  but where the drift function  $w(\cdot)$  is uncertain; we place a GP prior over the drift function  $w(u) \sim \mathcal{GP}(\mathbf{0}, K_\omega(u, u))$ . We thus define the monotonic warping process  $g_j(x)$  as the solution, at  $\tau = T$ , to the ODE:

$$g_j(x) := u_j(\tau = T; x) = \int_0^T w_j(u(\tau)) d\tau \quad (1)$$

subject to initial condition  $u(\tau = 0) := x$ . To draw a sample  $\mathbf{g}_j^{(s)}$  from the process we first draw a posterior *function* sample from the GP  $w^{(s)}(\cdot)$  and solve the resulting ODE jointly for all elements in  $\mathbf{x}_j$ . The use of an ODE rather than an SDE has the advantage of guaranteed smoothness, from the GP prior on  $w(\cdot)$ , and allows the use of higher-order adaptive solvers, *e.g.* Runge-Kutta (Schober et al., 2014).

The difficulty presented is the requirement to draw a single *function* sample from the GP for integration by the ODE solver. Typically, we draw joint samples from a GP posterior only for a given finite set of input locations. For the ODE, we do not know a priori all the input locations; they are only revealed sequentially as the solver progresses and depend on function evaluations for previous values of  $\tau$ . We solve this problem using a recent result in efficient path-wise sampling from GP posteriors from Wilson et al. (2020). This allows us, not only, to evaluate the sampled function  $w^{(s)}(\cdot)$  sequentially, but also to perform the evaluation efficiently without performing expensive Cholesky operations (which scale cubically with the number of posterior samples). Further details of this inference procedure are provided in Sec. 5.2. We note a concurrent work by Hegde et al. (2021) that similarly uses path-wise sampling from a GP to infer posterior of an ODE system.

**Joint Distribution** The joint probability distribution factorises as

$$p(\mathbf{y}, \mathbf{f}, \mathbf{z}, \mathbf{g}, w | \mathbf{X}, \beta, \theta, \psi) = p(\mathbf{f} | \mathbf{z}, \mathbf{g}, \theta, \psi) \prod_{j=1}^J p(\mathbf{g}_j | \mathbf{x}_j, w_j) p(w_j) p(\mathbf{z}_j) \prod_{n=1}^N p(y_{jn} | f_{jn}, \beta). \quad (2)$$

The terms in the joint distribution are:

$$\begin{aligned} \mathbf{y} | \mathbf{f}, \beta &\sim \mathcal{N}(\mathbf{y} | \mathbf{f}, \beta^{-1} \mathbf{I}_{JN}), \\ \mathbf{f} | \mathbf{z}, \mathbf{g} &\sim \mathcal{GP}(\mathbf{f} | \mathbf{0}, K_\psi(\mathbf{z}_j, \mathbf{z}_{j'}) \odot K_\theta(\mathbf{g}_{j,n}, \mathbf{g}_{j',n'})), \\ \mathbf{g}_j | \mathbf{x}_j, w_j &\sim \text{Monotonic Process}^2(\mathbf{g}_j | \mathbf{x}_j, w_j), \\ w_j &\sim \mathcal{GP}(w_j | \mathbf{0}, K_\omega(u_j, u_j)), \\ \mathbf{z}_j &\sim \mathcal{N}(\mathbf{z}_j | \mathbf{0}, \mathbf{I}_Q), \end{aligned} \quad (3)$$

where  $\odot$  denotes a tensor product such that  $[K_{\mathbf{f}, \mathbf{f}}]_{jn, j'n'} = [K_\psi(\mathbf{z}_j, \mathbf{z}_{j'})]_{j, j'} [K_\theta(\mathbf{g}_{j,n}, \mathbf{g}_{j',n'})]_{j, j'}$ . The functional values  $\mathbf{f}$  are fully correlated across all inputs and tasks that leads to the problematic complexity of  $\mathcal{O}(J^3 N^3)$ . Since all  $\{\mathbf{g}_j\}$  are now different, we can no longer utilise the Kronecker structure, as was suggested in previous work, *e.g.* Zhe et al. (2019). To address this we formulate a stochastic variational inference scheme, following the framework of Hensman et al. (2013).

## 5 Inference

Several parts of the model pose distinct challenges for inference. Firstly, the covariance of  $\mathbf{f}$  depends on both latent variables  $\mathbf{z}$  and the warps  $\mathbf{g}$ , hence we can not marginalize them out in closed form. Secondly, even if we use point estimates for  $\mathbf{z}$  and  $\mathbf{g}$  (*e.g.* MAP), the resulting covariance matrix would be of size  $JN \times JN$ , which is prohibitively expensive to invert. Notice, that we cannot use Kronecker decomposition of the covariance, a typical efficiency trick in multi-task GP models (Stegle et al., 2011; Dai et al., 2017), in the case of misaligned or missing data. To deal with these issues, and avoid point estimates, we adopt a sparse Stochastic Variational Inference (SVI) scheme.

We wish to compute the marginal likelihood of the data  $p(\mathbf{y} | \mathbf{X}) = \int p(\mathbf{y} | \mathbf{g}, \mathbf{z}) p(\mathbf{g} | \mathbf{X}) p(\mathbf{z}) d\mathbf{z} d\mathbf{g}$ . This integral is intractable as both latent variables  $\mathbf{z}$  and warps  $\mathbf{g}$  appear nonlinearly inside the inverse of the covariance matrix. To address this, we use a variational approach and introduce separable distributions over the latent variables  $q(\mathbf{z}) := \prod_{j=1}^J q(\mathbf{z}_j)$  and warps  $q(\mathbf{g}) := \prod_{j=1}^J q(\mathbf{g}_j)$  to approximate the true posterior  $p(\mathbf{z}, \mathbf{g} | \mathbf{y}, \mathbf{X})$ . The log marginal likelihood can then be bounded using Jensen’s inequality:

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}) &\geq \mathbb{E}_{q(\mathbf{z})q(\mathbf{g})} [\log p(\mathbf{y} | \mathbf{z}, \mathbf{g})] \\ &\quad - \text{KL}[q(\mathbf{z}) \| p(\mathbf{z})] - \text{KL}[q(\mathbf{g}) \| p(\mathbf{g})]. \end{aligned} \quad (4)$$

The expectation is still intractable, but we can further bound  $\mathcal{L}_1 := \log p(\mathbf{y} | \mathbf{z}, \mathbf{g})$  using the sparse VI approach of Titsias (2009).

<sup>2</sup>In the ODE formulation of Monotonic Process,  $\mathbf{g}_j | \mathbf{x}_j, w_j$  is deterministic, however, for simplicity, we often use  $\mathbf{g}_j$  to denote a general stochastic monotonic warp.

## 5.1 Sparse Stochastic Variational Inference

To make progress, we augment our model by introducing a set of inducing variables. Consider a set of  $M$  auxiliary variables  $\mathbf{h} \in \mathbb{R}^M$  evaluated at some artificial pseudo-inputs  $[\tilde{\mathbf{X}}, \tilde{\mathbf{Z}}]$ , where  $\tilde{\mathbf{X}} \in \mathbb{R}^M$  and  $\tilde{\mathbf{Z}} \in \mathbb{R}^{M \times Q}$ . We may then define an augmented joint distribution as

$$p(\mathbf{y}, \mathbf{f}, \mathbf{h}, \mathbf{z}, \mathbf{g} \mid \mathbf{X}, \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) = p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{h}, \mathbf{z}, \mathbf{g}, \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) p(\mathbf{h} \mid \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) p(\mathbf{g} \mid \mathbf{X}) p(\mathbf{z}). \quad (5)$$

Following the approach of (Titsias, 2009), we define a sparse approximation to the posterior distribution over  $\mathbf{f}$  using the inducing variables. Omitting the dependence on  $\mathbf{X}$  for clarity, the exact posterior over  $\mathbf{f}$  in the augmented model can be described by the predictive Gaussian distribution

$$p(\mathbf{f} \mid \mathbf{y}, \mathbf{g}, \mathbf{z}) = \int p(\mathbf{f} \mid \mathbf{h}, \mathbf{y}, \mathbf{g}, \mathbf{z}) p(\mathbf{h} \mid \mathbf{y}, \mathbf{g}, \mathbf{z}) d\mathbf{h}. \quad (6)$$

Suppose that  $\mathbf{h}$  is a sufficient statistic for  $\mathbf{f}$ , meaning that for any new inputs  $[\mathbf{X}^*, \mathbf{Z}^*]$  and the corresponding function values  $\mathbf{f}^*$ , we have  $\mathbf{f}^* \perp \mathbf{f} \mid \mathbf{h}$  or  $p(\mathbf{f}^* \mid \mathbf{h}, \mathbf{f}) = p(\mathbf{f}^* \mid \mathbf{h})$ . Similarly to Titsias (2009) (see supplement for more details), it follows that we can drop the dependence on  $\mathbf{y}$  in the posterior such that  $p(\mathbf{f} \mid \mathbf{h}, \mathbf{y}, \mathbf{g}, \mathbf{z}) = p(\mathbf{f} \mid \mathbf{h}, \mathbf{g}, \mathbf{z})$ . Under this assumption, we can write an approximation to the exact posterior in (6) as  $q(\mathbf{f}) = \int p(\mathbf{f} \mid \mathbf{h}, \mathbf{g}, \mathbf{z}) q(\mathbf{h}) d\mathbf{h}$ , where we specify that  $q(\mathbf{h}) := \mathcal{N}(\mathbf{h} \mid \mathbf{m}_h, \mathbf{S}_h)$ . The variational distribution over  $\mathbf{f}$  and  $\mathbf{h}$  is then  $q(\mathbf{f}, \mathbf{h}) = p(\mathbf{f} \mid \mathbf{h}, \mathbf{g}, \mathbf{z}) q(\mathbf{h})$ .

Now, using the augmented model and the variational distribution  $q(\mathbf{f}, \mathbf{h})$ , we can write the lower bound on  $\mathcal{L}_1$  as

$$\begin{aligned} \mathcal{L}_1 &= \log p(\mathbf{y} \mid \mathbf{z}, \mathbf{g}) \\ &= \log \int p(\mathbf{f} \mid \mathbf{h}, \mathbf{z}, \mathbf{g}) q(\mathbf{h}) \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{h})}{q(\mathbf{h})} d\mathbf{h} d\mathbf{f} \\ &\geq \mathcal{L}_2 - \text{KL}[q(\mathbf{h}) \parallel p(\mathbf{h})], \\ \mathcal{L}_2 &:= \int q(\mathbf{h}) \left[ \int p(\mathbf{f} \mid \mathbf{h}, \mathbf{g}, \mathbf{z}) \log p(\mathbf{y} \mid \mathbf{f}) d\mathbf{f} \right] d\mathbf{h}. \end{aligned} \quad (7)$$

While it is possible to ‘collapse’ the distribution  $q(\mathbf{h})$  by finding its optimal parameters (Titsias and Lawrence, 2010), we choose to follow the stochastic VI approach of Hensman et al. (2013) and keep the explicit representation of the inducing variables.

After marginalizing out  $\mathbf{f}$  and  $\mathbf{h}$  in the  $\mathcal{L}_2$  term of (7), please see the supplement for detailed derivation, the

overall lower bound  $\mathcal{L} \leq \log p(\mathbf{y})$  takes the form

$$\begin{aligned} \mathcal{L} &= \sum_{j=1}^J \left\{ \mathbb{E}_{q(\mathbf{z}_j)q(\mathbf{g}_j)} [\log \mathcal{N}(\mathbf{y}_j \mid \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1} \mathbf{m}, \beta^{-1} \mathbf{I})] \right. \\ &\quad - \frac{1}{2} \text{Tr}[\Lambda_j \mathbf{S}] - \frac{\beta}{2} \text{Tr}[\Sigma_j] - \text{KL}[q(\mathbf{z}_j) \parallel p(\mathbf{z}_j)] \\ &\quad \left. - \text{KL}[q(\mathbf{g}_j) \parallel p(\mathbf{g}_j)] \right\} - \text{KL}[q(\mathbf{h}) \parallel p(\mathbf{h})], \end{aligned} \quad (8)$$

where we have matrices  $\Lambda_j := \beta \mathbf{K}_{hh}^{-1} \mathbf{K}_{hf_j} \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1}$  and  $\Sigma_j := \mathbf{K}_{f_j, f_j} - \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1} \mathbf{K}_{hf_j}$ . The bound is factorised over sequences; combined with the separable kernels, we only need to compute the following expectations  $\mathbb{E}_{q(\mathbf{z}_i)} [K(\mathbf{z}_j, \tilde{\mathbf{Z}})]$ ,  $\mathbb{E}_{q(\mathbf{z}_i)} [K(\tilde{\mathbf{Z}}, \mathbf{z}_j) K(\mathbf{z}_j, \tilde{\mathbf{Z}})]$ ,  $\mathbb{E}_{q(\mathbf{g}_j)} [K(\mathbf{g}_j, \tilde{\mathbf{X}})]$  and  $\mathbb{E}_{q(\mathbf{g}_j)} [K(\tilde{\mathbf{X}}, \mathbf{g}_j) K(\mathbf{g}_j, \tilde{\mathbf{X}})]$ . In general, these can be approximated with sampling; the expectations under  $q(\mathbf{z})$  can be computed analytically for some kernels, *e.g.* the squared exponential.

Using sparse VI, the GP methodology allows for the Bayesian treatment of latent variables and warps, as well as reducing the time complexity to  $\mathcal{O}(JNM^2)$ . Stochastic VI adds the possibility of further complexity reduction through training using mini-batches of tasks.

## 5.2 Efficient SVI for Monotonic Warps

Calculation of (8) requires taking expectations over the warps under the approximate posteriors  $\{q(\mathbf{g}_j)\}$ . We estimate this bound by drawing samples from the respective monotonic processes. In Sec. 4.2, we outlined the sampling procedure as drawing a function sample from each GP posterior  $w_j^{(s)}(\cdot)$  and then solving the initial value ODE to obtain samples from  $q_j(\mathbf{g}_j)$ . As the inputs are unknown a priori, we follow Ustyuzhaninov et al. (2020) and Hegde et al. (2019), and specify the field using a sparse variational GP (Titsias, 2009). For each sequence, we define inducing locations  $\tilde{\mathbf{U}}_j$  and pseudo-outputs  $\mathbf{v}_j$ , and learn an approximate variational posterior  $q(\mathbf{v}_j) = \mathcal{N}(\mathbf{v}_j \mid \mathbf{m}_{w,j}, \mathbf{S}_{w,j})$ .

As the warps are smooth, we found it most efficient to solve the ODE using a simple Euler stepping approach with 10 steps over  $\tau \in [0, 1]$  taking gradients with respect to the variational parameters and kernel hyperparameters. The solver requires the sequential evaluation of a single functional sample  $w_j^{(s)}(\cdot)$  at arbitrary locations. Standard approaches would require all inputs to be known and a large covariance factorised. Instead, we make use of an efficient approximation scheme using path-wise samples from (Wilson et al., 2020). We combined Matheron’s Rule with a weight-space approximation to sample from the prior using random Fourier features (Rahimi and Recht,

2008). These samples may then be conditioned on the inducing-locations and samples from their corresponding pseudo-output distributions. Let  $\mathbf{\Omega}_j$  be a set of  $F$  random Fourier features for the kernel with hyperparameters  $\omega_j$  and  $\mathbf{b}_j$  be a set of draws from a uniform distribution over  $[0, 2\pi)$  such that  $\mathbf{\Omega}_j, \mathbf{b}_j \in \mathbb{R}^F$ . Then  $\phi_j(u) := \sqrt{2\sigma_\omega/F} \cos(\mathbf{\Omega}_j u + \mathbf{b}_j)$  defines a feature space such that  $K_{\omega_j}(\mathbf{u}, \mathbf{u}') \approx \phi_j^\top(\mathbf{u}) \phi_j(\mathbf{u}')$ . If we draw samples  $\boldsymbol{\alpha}^{(s)} \sim \mathcal{N}(\mathbf{0}, I_F)$  and  $\mathbf{v}_j^{(s)} \sim q(\mathbf{v}_j)$  then

$$w_j^{(s)}(u) := \phi_j^\top(u) \boldsymbol{\alpha}^{(s)} + \beta(u), \quad (9)$$

$$\beta(u) := K_{\omega_j}(u, \tilde{\mathbf{u}}_j) K_{\omega_j}^{-1}(\tilde{\mathbf{u}}_j, \tilde{\mathbf{u}}_j) (\mathbf{v}_j^{(s)} - \phi_j^\top(\tilde{\mathbf{u}}_j) \boldsymbol{\alpha}^{(s)})$$

gives a single functional draw of  $w_j^{(s)}(\cdot)$  for arbitrary  $u$ . Thus, fixing  $\boldsymbol{\alpha}^{(s)}$  and  $\mathbf{v}_j^{(s)}$  during the ODE solver loop, we efficiently solve for samples  $\mathbf{g}_j^{(s)}$  with complexity  $\mathcal{O}(N)$ .

When modelling monotonic warps  $\mathbf{g}$  as proposed, i.e. via ODE with variational GP drift, in eq. (8),  $q(\mathbf{g}_j)$  is replaced with  $p(w_j|\mathbf{v}_j)q(\mathbf{v}_j)$  and the corresponding KL divergence term becomes  $\text{KL}[q(\mathbf{v}_j) \| p(\mathbf{v}_j)]^3$ .

### 5.3 Learning

Training alternates two steps: (1) using natural gradients for the variational distributions of the inducing variables  $q(\mathbf{h}) = \mathcal{N}(\mathbf{h} | \mathbf{m}_h, \mathbf{S}_h)$  and  $q(\mathbf{v}_j) = \mathcal{N}(\mathbf{v}_j | \mathbf{m}_{w,j}, \mathbf{S}_{w,j})$  (see Hensman et al. (2013) for details); and (2) estimating  $q(\mathbf{z}_j) = \mathcal{N}(\mathbf{z}_j | \mathbf{m}_{z,j}, \text{diag}(\mathbf{s}_{z,j}))$  alongside the noise precision  $\beta$  and kernel hyperparameters  $\theta, \{\omega_j\}$  using the Adam optimizer (Kingma and Ba, 2014). We fix the latent space lengthscale and variance hyperparameters  $\psi$  to 1 to avoid excessive parameterization and initialise the latent variables  $\mathbf{z}$  using linear PCA. We use the GPflow framework (Matthews et al., 2017) and the GPflowSampling path sampling toolkit (Wilson et al., 2020). The code is available online<sup>4</sup>.

## 6 Experiments

To show that multi-task GP learning and inference benefits from alignment, we compare our AMTGP model against a version without the alignment functionality, denoted MTGP. MTGP can be seen as a fully Bayesian version of Latent Variable Multiple Output Gaussian Processes (Dai et al., 2017). To illustrate the benefits of marginalising out the warps, we also add results for the aligned model using point MAP estimates for the warps as in GP-LVA (denoted M-AMTGP). The MAP estimates are obtained by optimising a set

of auxiliary variables (constrained to be monotone) under a GP prior as proposed by GP-LVA (Kazlauskaitė et al., 2019). We also make comparison to GP-LVA.

We evaluate AMTGP on synthetic data as well as three real datasets: dynamic emotional facial expressions (Livingstone and Russo, 2018), heartbeat sounds (Bentley et al., 2011), and respiratory motion traces (Ernst, 2011) (please see the supplement). We perform a quantitative evaluation on the task of predicting missing data in three scenarios: (S1) data missing at random, (S2) a continuous segment of data missing at the same location for all tasks, and (S3) continuous segments of data missing at different locations for each task. The performance of the two approaches is compared using both the standardised mean squared error (SMSE) and the standardised negative log probability density (SNLP) (Williams and Rasmussen, 2006). The results are presented with statistics over 10 random data amputations. For all experiments, a Matérn 5/2 kernel is used for the warp differential field GP prior. For fair comparison, the number of inducing points for each model is chosen from the ELBO for the full dataset.

**Synthetic Data** We generate synthetic data by taking two 1-D functions and applying five random monotonic warps to each, adding i.i.d. Gaussian noise, to produce ten misaligned tasks. Missing data prediction performance is compared across the tasks in the three scenarios S1 - S3; 20% of the full data were removed and an SE kernel used. The results are summarised in Table 1 and Fig. 2. The latent variable posterior distribution (Figs. 2(a),(g)) shows that AMTGP and GP-LVA correctly identify the two underlying groups of tasks and that MTGP is unable to detect correlations between misaligned versions of the same task. The unaligned MTGP result (Fig. 2(c)) does not share information correctly and over-fits, resulting in large error bars and poor test performance. Whilst the M-AMTGP aligns the data and improves the mean, the point estimate of the warp is overconfident. With the full marginalisation of the warps, AMTGP is able to both align correctly and model the uncertainty accurately (Fig. 2(i)) resulting in improved performance for both SMSE and SNLP. Notably, in the scenarios with missing segments, S2 and S3, GP-LVA has very poor uncertainty estimation, confirming the detrimental effect of pseudo-observations on missing data reconstruction.

**Facial Expressions** We also test our method on a dataset of dynamic emotional facial expressions RAVDESS (Livingstone and Russo, 2018). This dataset contains recordings of people saying a short phrase with different emotions. We use mouth land-

<sup>3</sup>See B.3 in the supplement for details

<sup>4</sup>[https://github.com/01gaMikheeva/aligned\\_mtgp](https://github.com/01gaMikheeva/aligned_mtgp)

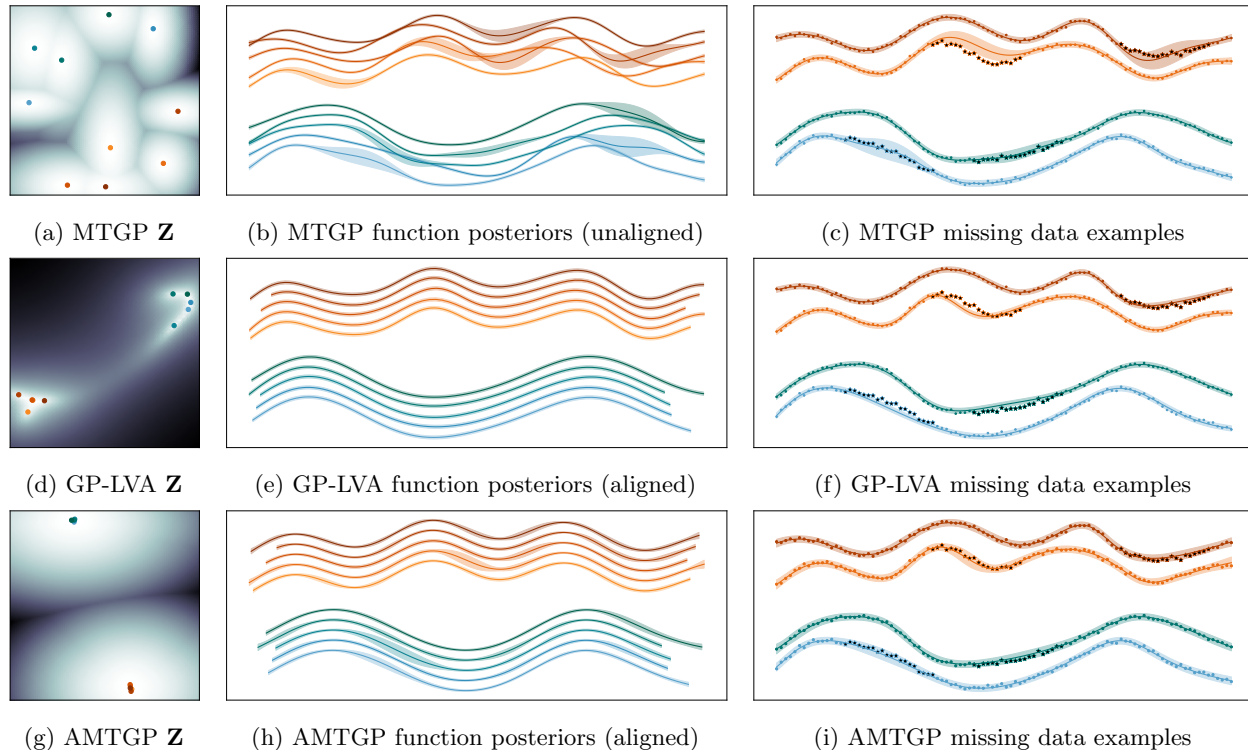


Figure 2: MTGP (top row), GP-LVA (middle row) and full AMTGP (bottom row) results on synthetic data for missing data scenario S3. (a), (d) and (g) show log-scaled posterior over the latent space. (b), (e) and (h) show posterior over  $f$  with  $2\sigma$  uncertainty bars. (g), (f) and (i) show the data and corresponding predictive distributions for two examples from the two different groups of tasks; this clustering (g) is correctly identified by AMTGP. Data points shown in black are the missing values and plots are vertically offset for clarity. MTGP has to introduce large error bars to account for the missing data (a) and, while it aligns correctly, GP-LVA becomes overconfident (f); in contrast, full AMTGP correctly accounts for the uncertainty in the warps and accurately models the missing data distribution (i).

Table 1: Results on the Synthetic Data.

	S1		S2		S3	
	Train	Test	Train	Test	Train	Test
MTGP (SMSE)	$0.0069 \pm 0.0003$	$0.0102 \pm 0.0009$	$0.0072 \pm 0.0006$	$0.1573 \pm 0.0624$	$0.0068 \pm 0.0002$	$0.157 \pm 0.0663$
GP-LVA (SMSE)	$0.0082 \pm 0.0004$	$0.0112 \pm 0.0008$	$0.0085 \pm 0.0009$	$1.3053 \pm 1.0803$	$0.0086 \pm 0.0024$	$0.1123 \pm 0.0727$
<b>M-AMTGP</b> (SMSE)	$0.0061 \pm 0.0002$	$0.0097 \pm 0.0009$	$0.0066 \pm 0.0006$	$0.0534 \pm 0.0181$	$0.0062 \pm 0.0001$	$0.0528 \pm 0.0235$
<b>AMTGP</b> (SMSE)	$0.0076 \pm 0.0002$	$0.0099 \pm 0.0007$	$0.0079 \pm 0.0007$	$0.052 \pm 0.0193$	$0.0076 \pm 0.0002$	$0.058 \pm 0.0235$
MTGP (SNLP)	$-1930.7 \pm 13.5$	$-456.4 \pm 7.6$	$-1919.8 \pm 32.4$	$-258.5 \pm 52.4$	$-1935.7 \pm 11.2$	$-167.1 \pm 72.4$
GP-LVA (SNLP)	$-1840.6 \pm 19.7$	$-399.7 \pm 16.0$	$-1806.4 \pm 41.4$	$14232.6 \pm 12776.9$	$-1810.5 \pm 74.5$	$1115.6 \pm 1126.2$
<b>M-AMTGP</b> (SNLP)	$-2024.9 \pm 10.3$	$-460.7 \pm 11.8$	$-1997.0 \pm 34.5$	$-181.6 \pm 128.3$	$-2015.7 \pm 8.1$	$-61.9 \pm 242.2$
<b>AMTGP</b> (SNLP)	$-1836.0 \pm 11.8$	$-442.0 \pm 7.4$	$-1826.6 \pm 36.6$	$-245.4 \pm 92.7$	$-1812.3 \pm 18.5$	$-156.3 \pm 141.3$

mark coordinate sequences extracted from the data, share warping functions across all coordinates from each recording, and use a Matérn5/2 kernel. We use two instances of the same phrase by the same person and ten mouth coordinates. Scenario S2 with 10% signal removal is employed on one instance and the other left intact. Both models are able to group lower and upper lip coordinates in each recording, but AMTGP also detects the similarity across recording instances, resulting in only two final clusters (Fig. 3(d)). The missing data prediction of AMTGP

is influenced by the behaviour of the other observed instance (Fig. 3(f)), while MTGP is unable to use this information resulting in phase errors (Fig. 3(c)).

**Heartbeat Sounds** We also consider sequences of heartbeat sounds recorded by a digital stethoscope (Bentley et al., 2011). A normal heart sound has a clear “lub dub, lub dub” pattern that varies temporally depending on the age, health, and state of the subject. The models are tested in scenario S1 with 20% missing data, a Matérn 5/2 kernel is used, and the



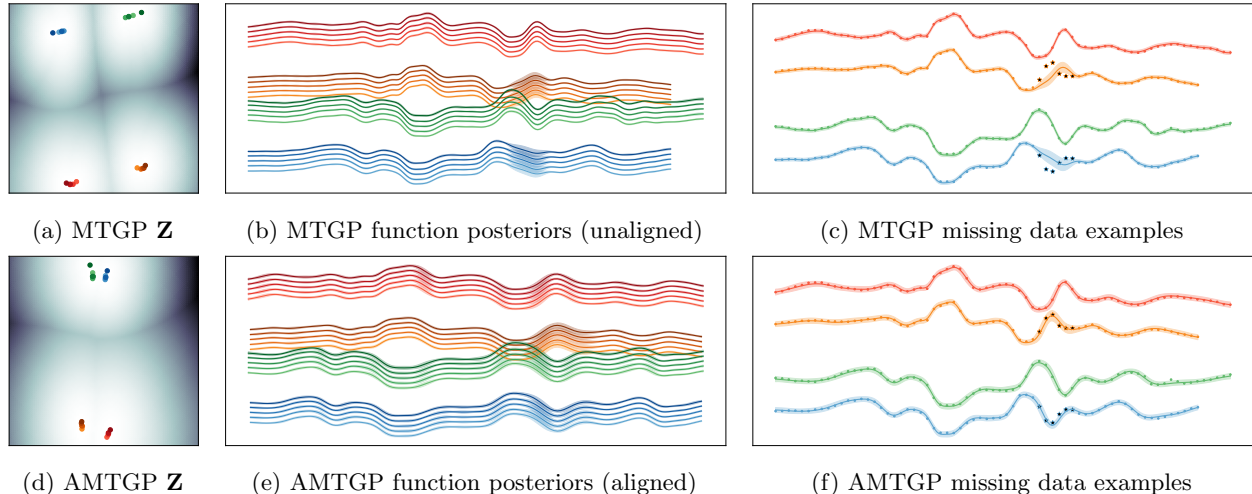


Figure 3: Experiment on the facial landmark data. Red and orange hues are the upper lip coordinates in 2 recordings, green and blue are the corresponding lower lip coordinates. Top row shows MTGP and the bottom row is AMTGP. (c) and (f) show one lower and one upper lip point for each recording. In contrast to AMTGP, the predictions under MTGP (c) are out of phase with the missing data.

Table 2: Results on Real Datasets

	Facial Expressions		Heartbeat Sounds	
	Train	Test	Train	Test
MTGP (SMSE)	$0.0027 \pm 0.0001$	$1.8327 \pm 2.9569$	$0.0329 \pm 0.0038$	$0.0544 \pm 0.0103$
GP-LVA (SMSE)	$0.0216 \pm 0.0021$	$0.5166 \pm 0.4626$	-	-
<b>M-AMTGP</b> (SMSE)	$0.0018 \pm 0.0001$	$0.8172 \pm 1.6459$	$0.0138 \pm 0.0005$	$0.0284 \pm 0.0138$
<b>AMTGP</b> (SMSE)	$0.0059 \pm 0.0002$	$0.6686 \pm 1.1964$	$0.0181 \pm 0.0014$	$0.0276 \pm 0.0046$
MTGP (SNLP)	$-3469.7 \pm 30.0$	$89.5 \pm 228.5$	$-1674.0 \pm 41.3$	$-378.4 \pm 25.1$
GP-LVA (SNLP)	$-2472.2 \pm 57.8$	$86.3 \pm 119.9$	-	-
<b>M-AMTGP</b> (SNLP)	$-3803.6 \pm 46.0$	$-44.5 \pm 35.4$	$-2231.1 \pm 21.2$	$-489.1 \pm 65.9$
<b>AMTGP</b> (SNLP)	$-2910.4 \pm 48.4$	$-56.4 \pm 42.3$	$-2006.6 \pm 36.5$	$-466.5 \pm 21.6$

results are presented in Table 2. As indicated by the latent space posterior (figures provided in the supplement), AMTGP uncovers correlations within groups of “lubs” and “dubs”, resulting in better predictive performance.

## 7 Conclusion and Limitations

AMTGP performs multi-task learning under GP priors that address the problem of temporal noise or warping in time-series data; it extends existing work on multi-task GPs to the case of warped inputs. We derive the variational bound, leveraging SVI and path-wise sampling for efficient fully Bayesian inference. We provide multiple examples to confirm the intuition that temporal alignment can and should be treated as an integral part of a multi-task model. We show, that while modelling uncertainty in the warps is not critical for alignment, it is beneficial when making predictions far

from existing data, when MAP can be over-confident. Monotonic warps are natural for time-series, however, other domains, *e.g.* spatial or image, other transformations may be more appropriate, *e.g.* rotations or translation. Our approach could be extended to infer the parameters of some other transformation function. Whilst modelling the warps adds a computational overhead, our efficient path-wise approach ensures a linear scaling in  $N$ . In practice, differentiating through the ODE solver is currently the main limiting factor. That said, the lack of Kronecker form also introduces extra complexity linearly in  $J$ . Those limitations will be considered in future work.

## Acknowledgements

This work was supported by the Swedish Foundation for Strategic Research, Grant No. RIT15-0107 (EACare), EPSRC grant EP/P020720/2 for Inference, CComputation and Numerics for Insights into Cities

(ICONIC, <https://iconicmath.org/>), the UKRI CAMERA project (EP/T022523/1) and the Royal Society. The authors would like to thank Akshaya Thip-pur for his feedback on the manuscript.

## References

- Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research (JMLR)*, 12, 2011.
- Mauricio A Álvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *arXiv preprint:1106.6251*, 2011.
- Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3), 2012.
- M. R. Andersen, E. Siivola, G. Riutort-Mayol, and A. Vehtari. A non-parametric probabilistic model for monotonic functions. “*All Of Bayesian Nonparametrics*” *Workshop at NeurIPS*, 2018.
- John Barnard, Robert McCulloch, and Xiao-Li Meng. Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, pages 1281–1311, 2000.
- P. Bentley, G. Nordehn, M. Coimbra, and S. Mannor. The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results. <http://www.peterjbentley.com/heartchallenge>, 2011.
- Ilias Bilonis, Nicholas Zabararas, Bledar A. Konomi, and Guang Lin. Multi-output separable Gaussian process: Towards an efficient, fully Bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241, 2013.
- Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.
- Ayman Boustati, Theodoros Damoulas, and Richard S Savage. Non-linear multitask learning with deep Gaussian processes. *arXiv preprint:1905.12407*, 2019.
- Phillip Boyle and Marcus Frean. Dependent Gaussian processes. *Advances in Neural Information Processing Systems (NeurIPS)*, 17, 2004.
- Zhenwen Dai, Mauricio A Álvarez, and Neil D Lawrence. Efficient modeling of latent information in supervised learning using gaussian processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5137–5145, 2017.
- Lea Duncker and Maneesh Sahani. Temporal alignment and latent gaussian process factor inference in population spike trains. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10466–10476, 2018.
- Robert Dürichen, Marco AF Pimentel, Lei Clifton, Achim Schweikard, and David A Clifton. Multitask Gaussian processes for multivariate physiological time-series analysis. *IEEE Transactions on Biomedical Engineering*, 62(1), 2014.
- Pierre Dutilleul. The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, 64(2), 1999.
- Floris Ernst. *Compensating for quasi-periodic motion in robotic radiosurgery*. Springer Science & Business Media, 2011.
- Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.
- Chiara Forlani, Samir Bhatt, Michela Cameletti, Elias Krainski, and Marta Blangiardo. A joint bayesian space-time model to integrate spatially misaligned air pollution data in r-inla. *Environmetrics*, 31(8): e2644, 2020.
- Pierre Goovaerts et al. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.
- Oliver Hamelijnck, Theodoros Damoulas, Kangrui Wang, and Mark Girolami. Multi-resolution multi-task Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- P. Hegde, M. Heinonen, H. Lähdesmäki, and S. Kaski. Deep learning with differential gaussian process flows. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.
- Pashupati Hegde, Çağatay Yıldız, Harri Lähdesmäki, Samuel Kaski, and Markus Heinonen. Bayesian inference of odes with gaussian processes. *arXiv preprint arXiv:2106.10905*, 2021.
- James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint:1309.6835*, 2013.
- AG Journel and CJ Huijbregts. *Mining Geostatistics*. Blackburn Press, 1978.
- Markus Kaiser, Clemens Otte, Thomas Runkler, and Carl Henrik Ek. Bayesian alignments of warped multi-output gaussian processes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6995–7004, 2018.

- Ieva Kazlauskaitė, Ivan Ustyuzhaninov, Carl Henrik Ek, and Neill DF Campbell. Sequence alignment with Dirichlet process mixtures. *arXiv preprint:1811.10689*, 2018.
- Ieva Kazlauskaitė, Carl Henrik Ek, and Neill Campbell. Gaussian process latent variable alignment learning. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- L. Lin and D.B. Dunson. Bayesian monotone regression using gaussian process projection. *Biometrika*, 101(2):303–317, 2014.
- Haitao Liu, Jianfei Cai, and Yew-Soon Ong. Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.
- Steven R Livingstone and Frank A Russo. The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS one*, 13(5):e0196391, 2018.
- A. F. Lopez-Lopera, ST John, and N. Durrande. Gaussian process modulated cox processes under linear inequality constraints. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.
- H. Maatouk. Finite-dimensional approximation of gaussian processes with inequality constraints. *arXiv:1706.02178*, 2017.
- Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrà, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017. URL <http://jmlr.org/papers/v18/16-537.html>.
- Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.
- Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 645–652. JMLR Workshop and Conference Proceedings, 2010.
- Sujit K Sahu and Kanti V Mardia. Recent trends in modelling spatio-temporal data. 2005.
- Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ODE solvers with Runge-Kutta means. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.
- Luigi Spezia. Modelling covariance matrices by the trigonometric separation strategy with application to hidden markov models. *Test*, 28(2):399–422, 2019.
- Oliver Stegle, Christoph Lippert, Joris M Mooij, Neil D. Lawrence, and Karsten Borgwardt. Efficient inference in matrix-variate Gaussian models with iid observation noise. *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.
- Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, 2009.
- Michalis Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.
- Ivan Ustyuzhaninov, Ieva Kazlauskaitė, Carl Henrik Ek, and Neill Campbell. Monotonic Gaussian process flows. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.
- Hans Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Verlag, 2003.
- Zheng Wang, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-fidelity high-order Gaussian processes for physical simulation. *arXiv preprint:2006.04972*, 2020.
- Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc P. Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.
- Shandian Zhe, Wei Xing, and Robert M Kirby. Scalable high-order Gaussian process regression. In *The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

---

## Supplementary Material: Aligned Multi-Task Gaussian Process

---

### A PGM

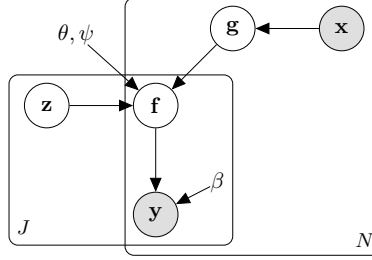


Figure 4: Generative model of **AMTGP**. The observed variables are shown in grey. The latent variables  $\mathbf{g}$  are the warps that map the inputs  $\mathbf{x}$  to the aligned input values. The latent variables  $\mathbf{z}$  encode the inter-task correlations.

### B Derivations

#### B.1 Sufficient Statistic Assumption

Following Titsias (2009) we can compute the predictive posterior at the new locations  $[\mathbf{X}^*, \mathbf{Z}^*]$  using the augmented joint:

$$\begin{aligned}
 p(\mathbf{f}^* | \mathbf{y}, \mathbf{g}, \mathbf{z}, \mathbf{X}^*, \mathbf{Z}^*) &= \int p(\mathbf{f}^*, \mathbf{f}, \mathbf{h} | \mathbf{y}, \mathbf{g}, \mathbf{z}, \mathbf{X}^*, \mathbf{Z}^*) d\mathbf{f} d\mathbf{h} \\
 &= \int p(\mathbf{f}^* | \mathbf{f}, \mathbf{h}, \mathbf{g}, \mathbf{z}, \mathbf{X}^*, \mathbf{Z}^*) p(\mathbf{f} | \mathbf{h}, \mathbf{y}, \mathbf{g}, \mathbf{z}) p(\mathbf{h} | \mathbf{y}, \mathbf{g}, \mathbf{z}) d\mathbf{f} d\mathbf{h}
 \end{aligned} \tag{10}$$

Let us assume that  $\mathbf{h}$  is a sufficient statistic for  $\mathbf{f}$ , meaning:  $\mathbf{f}^* \perp \mathbf{f} | \mathbf{h}$  or  $p(\mathbf{f}^* | \mathbf{h}, \mathbf{f}) = p(\mathbf{f}^* | \mathbf{h})$ . Thus we have,  $p(\mathbf{f}^* | \mathbf{f}, \mathbf{h}, \mathbf{g}, \mathbf{z}, \mathbf{X}^*, \mathbf{Z}^*) = p(\mathbf{f}^* | \mathbf{h}, \mathbf{g}, \mathbf{z}, \mathbf{X}^*, \mathbf{Z}^*)$ .

Due to the assumption and the fact that  $\mathbf{y}$  is a noisy version of  $\mathbf{f}$ , it follows that

$$\begin{aligned}
 p(\mathbf{f}^* | \mathbf{h}, \mathbf{y}, \mathbf{g}, \mathbf{z}) &= \frac{\int p(\mathbf{f}^*, \mathbf{f}, \mathbf{h}, \mathbf{y} | \mathbf{g}, \mathbf{z}) d\mathbf{f}}{\int p(\mathbf{f}^*, \mathbf{f}, \mathbf{h}, \mathbf{y} | \mathbf{g}, \mathbf{z}) d\mathbf{f} d\mathbf{f}^*} \\
 &= \frac{\int p(\mathbf{y} | \mathbf{f}) p(\mathbf{f}^*, \mathbf{f}, \mathbf{h} | \mathbf{g}, \mathbf{z}) d\mathbf{f}}{\int p(\mathbf{y} | \mathbf{f}) p(\mathbf{f}^*, \mathbf{f}, \mathbf{h} | \mathbf{g}, \mathbf{z}) d\mathbf{f} d\mathbf{f}^*} \\
 &= \frac{\int p(\mathbf{y} | \mathbf{f}) p(\mathbf{f}^* | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{h} | \mathbf{g}, \mathbf{z}) d\mathbf{f}}{\int p(\mathbf{y} | \mathbf{f}) p(\mathbf{f}^* | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{h} | \mathbf{g}, \mathbf{z}) d\mathbf{f} d\mathbf{f}^*} \\
 &= \frac{p(\mathbf{y} | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{f}^*, \mathbf{h} | \mathbf{g}, \mathbf{z})}{p(\mathbf{y} | \mathbf{h}, \mathbf{g}, \mathbf{z}) p(\mathbf{h} | \mathbf{g}, \mathbf{z})} = p(\mathbf{f}^* | \mathbf{h}, \mathbf{g}, \mathbf{z})
 \end{aligned} \tag{11}$$

Hence,  $p(\mathbf{f} | \mathbf{h}, \mathbf{y}, \mathbf{g}, \mathbf{z}) = p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z})$ .

## B.2 $\mathcal{L}_2$ Bound

$\mathcal{L}_2$  is defined as  $\mathcal{L}_2 = \int q(\mathbf{h})p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z}) \log p(\mathbf{y} | \mathbf{f}) d\mathbf{f} d\mathbf{h}$ , where  $q(\mathbf{h}) := \mathcal{N}(\mathbf{h} | \mathbf{m}, \mathbf{S})$ . Marginalising out  $\mathbf{f}$  we get

$$\mathcal{L}_2 = \int q(\mathbf{h})p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z}) \log p(\mathbf{y} | \mathbf{f}) d\mathbf{f} d\mathbf{h} \quad (12)$$

$$= \mathbb{E}_{q(\mathbf{h})} \left[ \int p(\mathbf{f} | \mathbf{h}, \mathbf{g}, \mathbf{z}) \log p(\mathbf{y} | \mathbf{f}) d\mathbf{f} \right] \quad (13)$$

$$= \mathbb{E}_{q(\mathbf{h})} \left[ -\frac{\beta}{2}(\mathbf{y} - \boldsymbol{\mu})^T(\mathbf{y} - \boldsymbol{\mu}) - \frac{JN}{2} \log 2\pi - \frac{1}{2} \log |\beta^{-1}\mathbf{I}| - \frac{\beta}{2} \text{Tr}[\boldsymbol{\Sigma}] \right] \quad (14)$$

where  $\boldsymbol{\mu} := \mathbf{K}_{fh}\mathbf{K}_{hh}^{-1}\mathbf{h}$  and  $\boldsymbol{\Sigma} := \mathbf{K}_{ff} - \mathbf{K}_{fh}\mathbf{K}_{hh}^{-1}\mathbf{K}_{hf}$ .

Taking expectations under the distribution  $q(\mathbf{h})$  we arrive at the final result:

$$\mathcal{L}_2 = \sum_{j=1}^J \left\{ \log \mathcal{N}(\mathbf{y}_j | \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1} \mathbf{m}, \beta^{-1} \mathbf{I}) - \frac{1}{2} \text{Tr}[\Lambda_j \mathbf{S}] - \frac{\beta}{2} \text{Tr}[\Sigma_j] \right\} \quad (15)$$

where the matrices  $\Lambda_j := \beta \mathbf{K}_{hh}^{-1} \mathbf{K}_{hf_j} \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1}$  and  $\Sigma_j := \mathbf{K}_{f_j, f_j} - \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1} \mathbf{K}_{hf_j}$ .

## B.3 Bound for Monotonic GP warps

Here we provide the derivation of the bound when modelling monotonic warps via ODE with variational drift GP. In this case,  $\mathbf{g}_j = g_j(w_j, \mathbf{x}_j)$ . Joint augmented distribution is

$$p(\mathbf{y}, \mathbf{f}, \mathbf{h}, \mathbf{z}, w, \mathbf{v} | \mathbf{X}) = p(\mathbf{f} | \mathbf{h}, \mathbf{z}, g(w, \mathbf{X})) p(\mathbf{h}) \prod_{j=1}^J p(w_j | \mathbf{v}_j) p(\mathbf{v}_j) p(\mathbf{z}_j) \prod_{n=1}^N p(y_{jn} | f_{jn}) \quad (16)$$

The approximate posterior is  $q(\mathbf{f}, \mathbf{h}, \mathbf{z}, w, \mathbf{v}) = p(\mathbf{f} | \mathbf{h}, \mathbf{z}, g(w, \mathbf{X})) q(\mathbf{h}) \prod_{j=1}^J p(w_j | \mathbf{v}_j) q(\mathbf{v}_j) q(\mathbf{z}_j)$ .

ELBO is then derived as follows

$$\log p(\mathbf{y} | \mathbf{X}) \geq \mathbb{E}_{q(\mathbf{f}, \mathbf{h}, \mathbf{z}, w, \mathbf{v})} \left[ \log \frac{p(\mathbf{f} | \mathbf{h}, \mathbf{z}, g(w, \mathbf{X})) p(\mathbf{h}) \prod_{j=1}^J p(w_j | \mathbf{v}_j) p(\mathbf{v}_j) p(\mathbf{z}_j) \prod_{n=1}^N p(y_{jn} | f_{jn})}{p(\mathbf{f} | \mathbf{h}, \mathbf{z}, g(w, \mathbf{X})) q(\mathbf{h}) \prod_{j=1}^J p(w_j | \mathbf{v}_j) q(\mathbf{v}_j) q(\mathbf{z}_j)} \right] \quad (17)$$

$$= \sum_{j=1}^J \left( \mathbb{E}_{q(\mathbf{f}, \mathbf{z}, w, \mathbf{v})} [\log p(\mathbf{y}_j | \mathbf{f}_j)] - \text{KL}[q(\mathbf{v}_j) \| p(\mathbf{v}_j)] - \text{KL}[q(\mathbf{z}_j) \| p(\mathbf{z}_j)] \right) - \text{KL}[q(\mathbf{h}) \| p(\mathbf{h})] \quad (18)$$

$$= \sum_{j=1}^J \left( \mathbb{E}_{q(\mathbf{z}_j) p(w_j | \mathbf{v}_j) q(\mathbf{v}_j)} [\log \mathcal{N}(\mathbf{y}_j | \mathbf{K}_{f_j h} \mathbf{K}_{hh}^{-1} \mathbf{m}, \beta^{-1} \mathbf{I}) - \frac{1}{2} \text{Tr}[\Lambda_j \mathbf{S}] - \frac{\beta}{2} \text{Tr}[\Sigma_j]] \right) \quad (19)$$

$$- \text{KL}[q(\mathbf{v}_j) \| p(\mathbf{v}_j)] - \text{KL}[q(\mathbf{z}_j) \| p(\mathbf{z}_j)] \Big) - \text{KL}[q(\mathbf{h}) \| p(\mathbf{h})] \quad (20)$$

## B.4 M-AMTGP: MAP estimate of the warps

For comparison, we include a version of AMTGP where we consider only a point MAP estimate for the monotonic warps following the GP-LVA approach (Kazlauskaitė et al., 2019).

We parameterise each  $\mathbf{g}_j$  through a set of auxiliary variables  $\mathbf{e}_j \in \mathbb{R}^N$  and enforce monotonicity as follows:

$$g_{jn} = \sum_{i=1}^n [\text{softmax}(\mathbf{e}_j)]_i, \quad (21)$$

under a Gaussian process prior over the values of  $\mathbf{g}_j$ . This re-parameterization ensures that the warps are monotonic over the specified range  $[0, 1]$ . We also add a scale and a shift parameter to map this to the range of the input space.

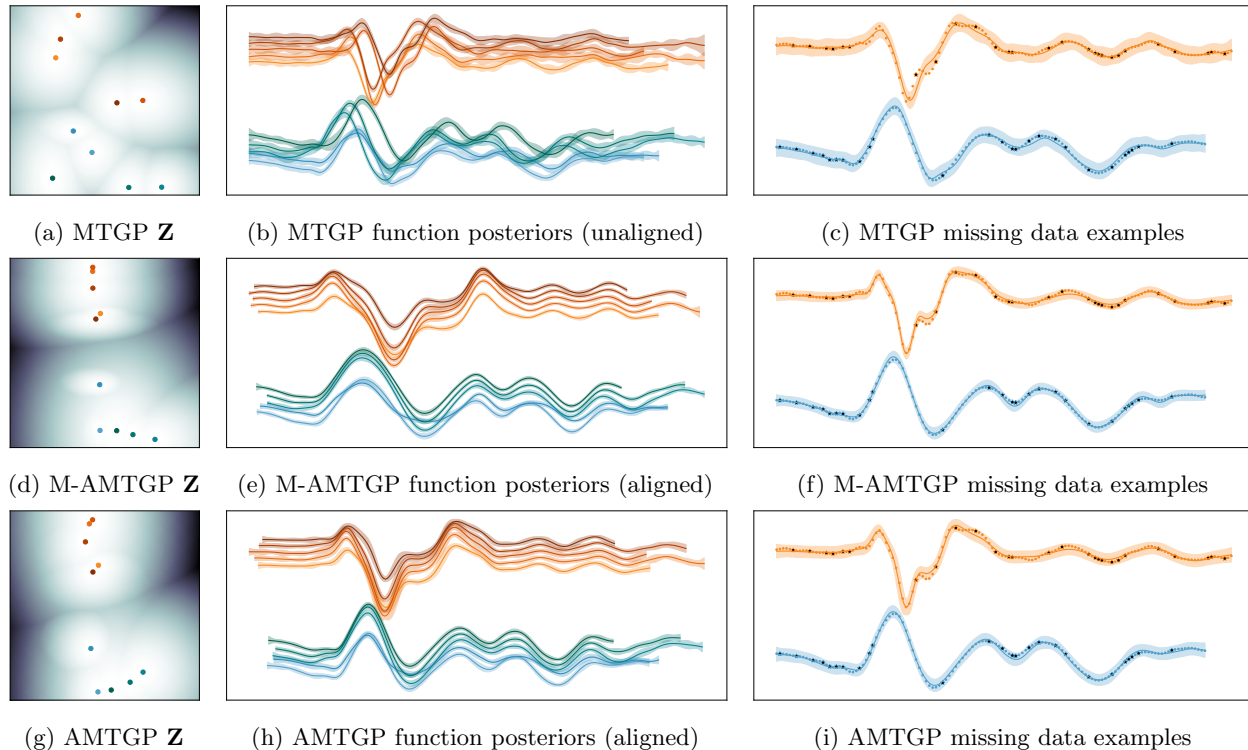


Figure 5: Experiment on the heart beat data. Top row shows MTGP, middle row is M-AMTGP (AMTGP with MAP estimate of warps), bottom row is AMTGP. “Lubs” and “dubs” are plotted with red and blue respectively. The right column shows one “lub” and one “dub”. 20% of the data is missing at random locations and is shown in black.

## B.5 GP-LVA: extension to missing data

The original GP-LVA (Kazlauskaitė et al., 2019) was not designed for missing data reconstruction. However, it is possible to use predictive posterior of individual sequences conditioned on both observed data  $Y$  and pseudo-observations  $S$ . While conditioning on  $S$  allows for implicit correlation between aligned sequences, it can cause the model severely misbehave. In some scenarios, we found the pseudo-observations to overpower the data, resulting in poor fit to  $Y$ .

## C Additional Experiments and Plots

### C.1 Heartbeat Data

Figure 5 shows results on the Heartbeat data. Details about the data are presented in the main paper. In this experiment, we omit the results for GP-LVA, since we could not achieve a good fit from this model even with no data missing. In particular, GP-LVA aligned pseudo-observations of both “lubs” and “dubs” together in one single category. This resulted in GP-LVA fitting the data  $Y$  well only on one category and severely misfitting on another. This clearly shows the downside of pseudo-observations trick.

Figure 6 shows posterior warps on the heartbeat experiment (the shift is removed, one of the warps is set to identity mean). As we can see, the necessary alignment is more complex than a simple shift or scale. This illustrates the need to flexibly model temporal warps in the real data.

### C.2 Respiratory Motion Traces

We perform additional experiments on physiological data (Ernst, 2011). Specifically, we consider recordings of human liver motion traces. We use 6 markers of liver positions (3 external data and 3 from ultrasound). In this data the misalignment is small and comes from physiological processes.

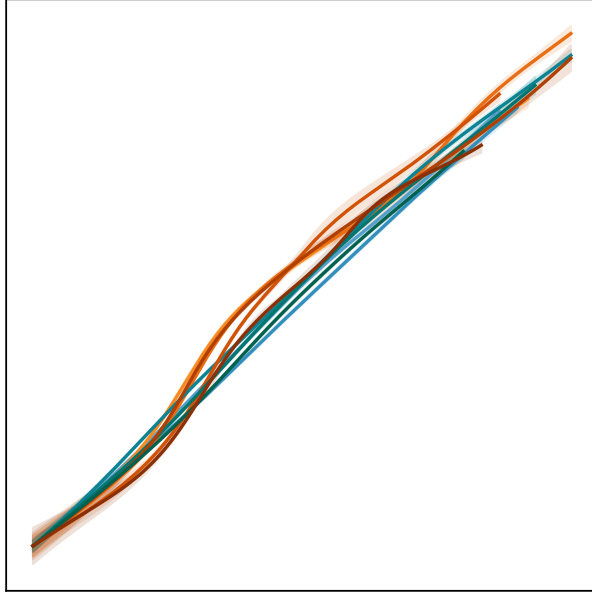


Figure 6: Posterior warps on heartbeat data.

Table 3: Results of the additional experiments.

	Respiratory motion traces (liver)		Facial Expressions 2	
	Train	Test	Train	Test
MTGP (SMSE)	$0.0084 \pm 0.0047$	$0.1787 \pm 0.1443$	$0.0726 \pm 0.006$	$0.2681 \pm 0.0997$
GP-LVA (SMSE)	$0.0362 \pm 0.0024$	$0.3075 \pm 0.3214$	$0.0479 \pm 0.0088$	$0.1804 \pm 0.0614$
<b>M-AMTGP</b> (SMSE)	$0.001 \pm 0.0005$	$0.1663 \pm 0.1370$	$0.0436 \pm 0.0072$	$0.1785 \pm 0.0753$
<b>AMTGP</b> (SMSE)	$0.0083 \pm 0.0021$	$0.1919 \pm 0.1533$	$0.054 \pm 0.009$	$0.1666 \pm 0.0941$
MTGP (SNLP)	$-1215.1 \pm 95.1$	$-71.8 \pm 22.9$	$-1885.2 \pm 55.0$	$-100.7 \pm 41.3$
GP-LVA (SNLP)	$-916.7 \pm 20.0$	$83.8 \pm 108.3$	$-2445.9 \pm 106.2$	$71.5 \pm 41.6$
<b>M-AMTGP</b> (SNLP)	$-1788.2 \pm 98.1$	$435.7 \pm 536.3$	$-2310.2 \pm 107.6$	$-112.8 \pm 60.3$
<b>AMTGP</b> (SNLP)	$-1210.4 \pm 59.5$	$-43.6 \pm 46.1$	$-2031.6 \pm 101.4$	$-131.1 \pm 63.1$

We perform an experiment where 10% of the data is missing in S3 scenario (continuous segment of data is missing in each task at random locations). Since the data clearly exhibits periodic behavior, we use a sum of a cosine and a squared exponential kernel to model the temporal behavior.

Results are presented in Table 3 and Figure 7. As we can see, all 3 models show similar predictive performance in terms of SMSE. This is due to the fact, that the data has only small misalignment and the correlated tasks exhibit strong periodic behaviour. Importantly, alignments in the proposed AMTGP model and its MAP version, M-AMTGP, do not lead to spurious correlations and negative knowledge transfer between the tasks. The predictive log probability (SNLP) in this experiment once again highlights the effect of Bayesian warp estimation, compared to MAP estimation. M-AMTGP is overconfident in its predictions, while fully Bayesian AMTGP preserves reasonable prediction uncertainty.

### C.3 Extra Experiment with Facial Expression Data

Here we present an additional experiment on the facial expressions data. For this experiment, we consider two mouth coordinates (lower and upper lip) and 14 recording instances yielding 28 tasks. We learn one warping function per instance and use scenario S3 (continuous segment of data missing at random location for each task) with 10% of the data removed.

The results of the experiment are presented in Table 3 and Figure 8. We can see, that unlike MTGP, both

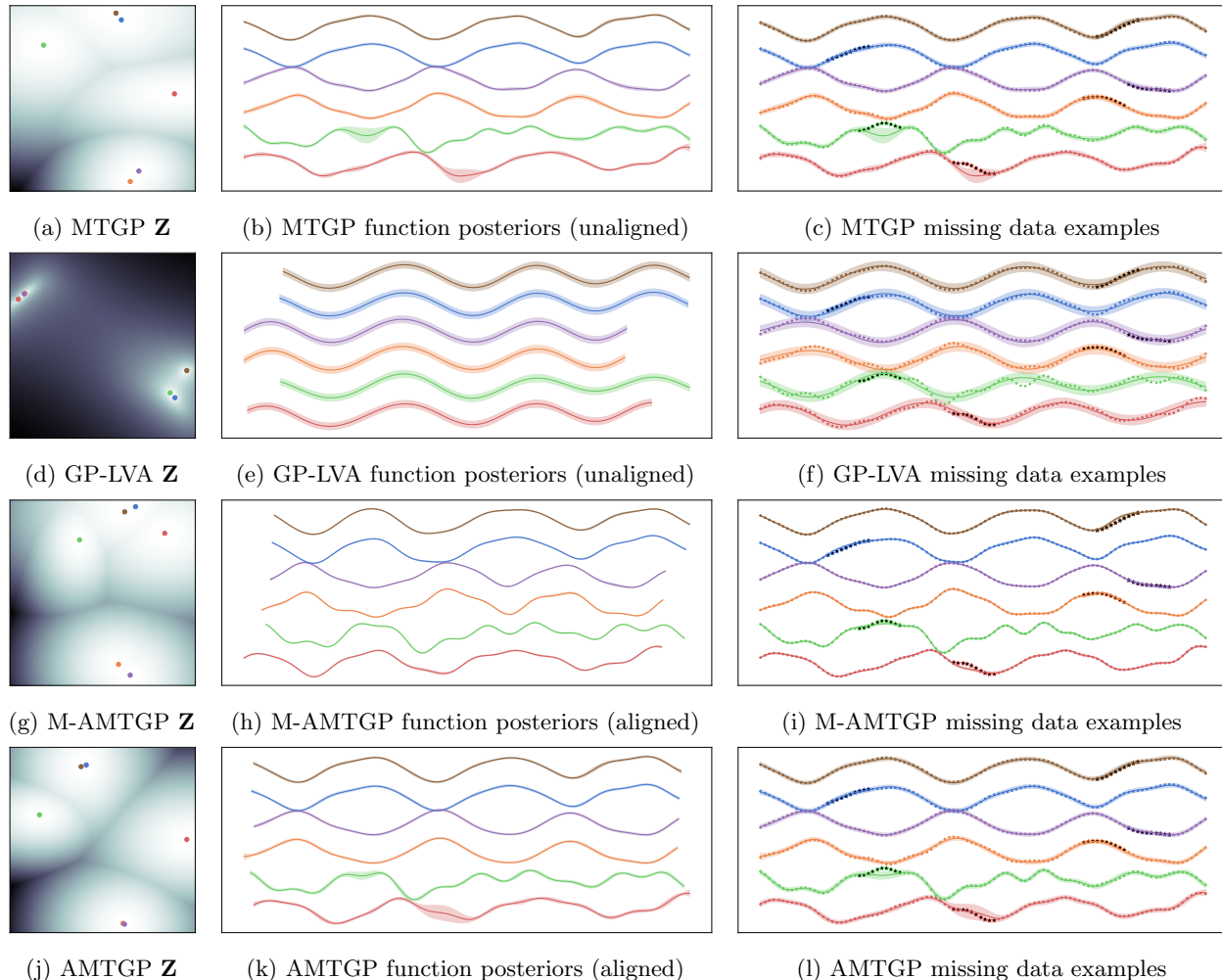


Figure 7: Experiment on the liver data. Top row shows MTGP, middle row is M-AMTGP (AMTGP with MAP estimate of warps), bottom row is AMTGP. The right column is plotted with an offset for better visibility. 10% of the data is missing in continuous segments at random locations in each task and is shown in black.

AMTGP and M-AMTGP are able to align and group lower and upper lip coordinated. This is reflected in better predictive performance (table 3) of the aligned models, with fully Bayesian AMTGP being the top model.

#### C.4 Sequence Alignment

Our model can be used not only for misaligned multi-task learning, but also for sequence alignment. In this aspect, our model, Aligned-MTGP can be seen as a more probabilistically solid formulation and generalization of the GP-LVA model (Kazlauskaitė et al., 2019). Here we compare the quality of alignments between GP-LVA, M-AMTGP and AMTGP on four synthetic datasets with known warps from Kazlauskaitė et al. (2019). Since GP-LVA and M-AMTGP only give point estimate of the warps, we use posterior warp means in AMTGP for comparison. As the task of alignment is underdetermined, to compare the warps we consider relative warps within each group (one true underlying function). In Table 4, we report the MSE between the true warps and the estimated warps with statistics over different reference warp choices. The identity warps (*i.e.* unaligned) are shown as a baseline. We can see that our method uncovers the true warps on par with GP-LVA while also providing a rigorous formulation of the model and the bound for the inference.

The synthetic datasets for testing warps have 10 sequences each. Generating functions are shown in Table 4.



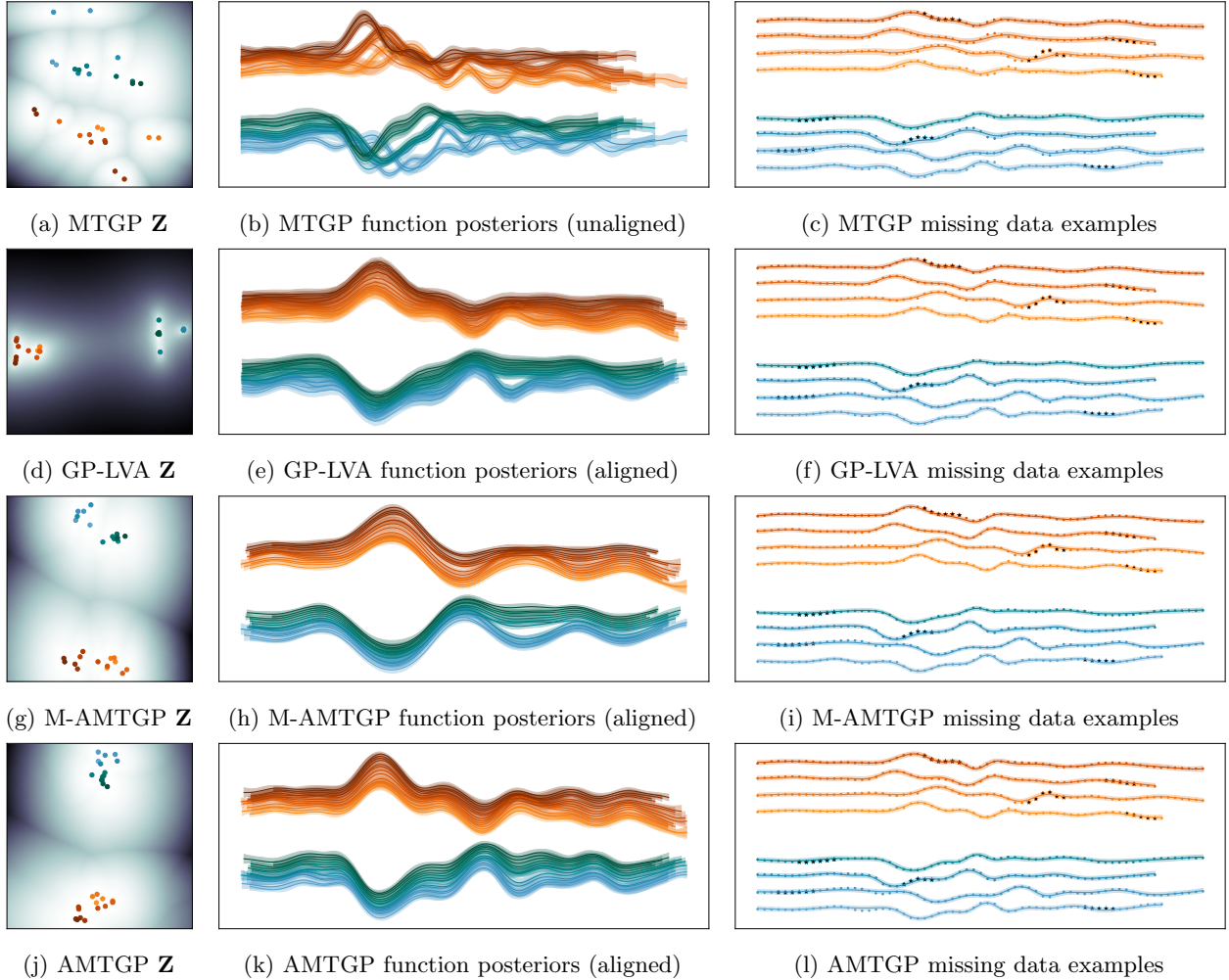


Figure 8: Experiment 2 on the facial landmark data. Top row shows MTGP, second row is GP-LVA, third row is M-AMTGP (AMTGP with MAP estimate of warps), bottom row is AMTGP. (a), (d), (g) and (j) show the latent space, (b), (e), (h) and (k) show the posterior over the aligned functions. Data fit is illustrated in (c), (f), (i) and (l) for a few tasks. 10% of the data is missing and is shown in black. Offset and colour coding separates upper and lower lip coordinates.

Table 4: Warp Recovery on Synthetic Data.

	Identity Warps	GP-LVA	M-AMTGP (Ours)	AMTGP (Ours)	Functions
1	$0.061 \pm 0.09$	$0.0042 \pm 0.0070$	$0.0017 \pm 0.0026$	$0.0050 \pm 0.0064$	$\text{sinc}(\pi x); 0.6x^3$ (5/5)
2	$0.061 \pm 0.09$	$0.0025 \pm 0.0036$	$0.0024 \pm 0.0036$	$0.0046 \pm 0.0081$	$\sin(\pi x); 0.6x^3$ (5/5)
3	$0.035 \pm 0.03$	$0.0076 \pm 0.0010$	$0.0055 \pm 0.0088$	$0.0052 \pm 0.0086$	$\sin(3x); 0.6x^3$ (5/5)
4	$0.080 \pm 0.09$	$0.0006 \pm 0.0008$	$0.0004 \pm 0.0004$	$0.0006 \pm 0.0011$	$\sin(6x)$ (10)

## D Model Parameters and Implementation

### D.1 Model and Training Parameters

For all experiments, the number of inducing points for each model is chosen based on the saturation of ELBO for the full dataset (see Table 5). This ensures that the performance is not affected by the quality of sparse approximation. As a result, the aligned models (AMTGP and M-AMTGP) require fewer inducing points.

For the inference over monotonic warps in AMTGP model we use 10 inducing points on a fixed grid for each warp function.

The details about the size of the dataset used in each experiment is provided in Table 6.

Modeling choices are summarized in Table 7.

For optimization we use a combination of Adam and natural gradients. We use natural gradients for  $q(h)$  with  $\gamma_h = 0.5$  for all models and experiments. While it is possible to use natural gradients for parameters of  $q(w)$  (underlying GPs in warps), we find that  $\gamma_w$  in this case has to be very low and does not improve convergence compared to Adam for those parameters. In synthetic data experiments, we use natural gradient for the warps with  $\gamma_w = 0.05$ , in real data experiments we use Adam.

The number of the training iterations for each experiment was determined based on the loss convergence and is reported in Table 7.

For all experiments, we train all models on 10 random data amputations and report mean and standard deviation of SMSE and SNLP.

Table 5: Number of Inducing Points.

	MTGP	M-AMTGP	AMTGP	GP-LVA (per task)
Synthetic data	250	50	50	25
Facial Expressions	200	200	100	20
Heartbeat Sounds	200	100	100	-
Facial Expressions 2	250	50	50	20
Respiratory Motion Traces	100	100	100	10

Table 6: Dataset size.

	# tasks	Sequence length	Total # of data points
Synthetic data	10	100	1000
Facial Expressions	20	65 - 70	1350
Heartbeat Sounds	10	120 - 150	1368
Facial Expressions 2	28	56 - 70	1704
Respiratory Motion Traces	6	100	600

Table 7: Model parameters and training details.

	Temporal kernel	# warp functions	Warp prior in M-AMTGP	Training iterations
Synthetic data	SE	10	SE(0.1, 0.1)	2000
Facial Expressions	Matern5/2	2	SE(0.1, 0.1)	2000
Heartbeat Sounds	Cosine + SE	10	SE(0.1, 1.)	3000
Facial Expressions 2	Matern5/2	14	SE(0.1, 0.1)	2000
Respiratory Motion Traces	Matern5/2	6	SE(0.1, 0.01)	3000

## D.2 Implementation Details

The model is implemented in Tensorflow. We make use of the GPflow framework (Matthews et al., 2017) and the GPflowSampling path sampling toolkit (Wilson et al., 2020).

The code is available at [https://github.com/OlgaMikheeva/aligned\\_mtgp](https://github.com/OlgaMikheeva/aligned_mtgp).

## D.3 Computational time

Approximate computational time for each experiment is provided in Table 8. The estimate is provided for a MacBook Pro with 2,7 GHz Quad-Core Intel Core i7 and 16 GB memory.

Higher computational time of AMTGP comes from the ODE solver. While extra evaluations of the sampled functions within the solver add to the complexity, most of the computational overhead comes from differentiating through the solver. We illustrate this in Table 9.

Table 8: Approximate computational time, in minutes

	<b>GP-LVA</b>	<b>MTGP</b>	<b>M-AMTGP</b>	<b>AMTGP</b>
Synthetic data		1	5	5 60
Facial Expressions		1	5	5 60
Heartbeat Sounds		1	5	5 60
Facial Expressions		1	5	5 60
Respiratory Motion Traces		1	5	5 60

Table 9: Analysis of computational time of AMTGP (average time per iteration, sec.)

# samples	# Fourier Features	Pathwise GP sampling	Pathwise monotonic GP sampling	Backprop
1	1024	0.00102	0.00606	0.03842
1	256	0.00077	0.00442	0.02084
10	1024	0.00120	0.05216	0.38295
10	256	0.00080	0.01284	0.08918
100	1024	0.00190	0.68137	3.87087
100	256	0.00104	0.15722	0.93967