Generalised Gaussian Process Latent Variable Models with Stochastic Variational Inference

Vidhi Lalchand, Aditya Ravuri, Neil D. Lawrence
University of Cambridge

Abstract

Gaussian process latent variable models (GPLVM) are a flexible and non-linear approach to dimensionality reduction [Lawrence, 2004], through a classical unsupervised learning paradigm. The Bayesian incarnation of the GPLVM uses a variational framework, where the posterior over latent variables is approximated by a well-behaved variational family, a factorised Gaussian [Titsias and Lawrence, 2010] yielding a tractable lower bound. However, the non-factorisability of the lower bound prevents truly scalable inference. This paper has three main contributions. (1) We recast the Bayesian GPLVM model to derive a *doubly stochastic evidence lower bound* amenable to stochastic variational inference (SVI) in the latent variable setting. (2) We exploit the SVI framework to allow training of the Bayesian GPLVM when over half the data is missing. (3) We amortise variational inference with an encoder that retains probabilistic representations in latent space. We demonstrate the model's performance by benchmarking against the canonical sparse GPLVM for high dimensional data examples.

1 Introduction

Gaussian processes (GPs) represent a powerful non-parametric probabilistic framework for performing regression and classification. The inductive biases are controlled by a kernel function [Rasmussen and Williams, 2006]. The Gaussian process latent variable model (GPLVM) [Lawrence, 2004] paved the way for GPs to be used in unsupervised learning tasks like dimensionality reduction and structure discovery for high-dimensional data. It provides a probabilistic mapping from (an unobserved) latent space ($\mathbf{X}$) to data-space ($\mathbf{Y}$). The GP acts as a *decoder*; the smoothness of the mapping is controlled by a kernel function. Many traditional dimensionality reduction models learn a projection of high dimensional data to lower dimensional manifolds [e.g. Jolliffe, 1986; Roweis and Saul, 2000]. In the GPLVM the direction of the mapping is reversed.

The standard GPLVM is a multi-output regression model whose the inputs are unobserved during training. The canonical formulation treats the unknown latent variables as point estimates and optimizes the marginal likelihood jointly with the covariance hyperparameters ($\boldsymbol{\theta}$). Techniques to apply Gaussian processes to very large datasets were introduced in [Hensman et al., 2013] which demonstrated how stochastic variational inference (SVI) [Hoffman et al., 2013] can be used with sparse GPs in a regression context. The key idea is to re-formulate the evidence lower bound (ELBO) Titsias [2009] in a way that factorizes across the data enabling mini-batching for gradients. The canonical formulation can be made sparse by using the regression based lower bound from [Hensman et al., 2013] and optimising for latents $\mathbf{X}$. We call this model the *Sparse GPLVM* and POINT for short. We also study the performance of maximum-a-posteriori (MAP) in this framework.

The Bayesian formulation of the GPLVM in [Titsias and Lawrence, 2010] variationally integrates out latent variables, providing principled uncertainty around the latent encoding. The sparse variational formulation relies on inducing variables [Titsias, 2009] that admit a tractable lower bound while providing computational savings. The Bayesian formulation also allows the dimensionality of the

Preprint. Under review.

Table 1: Existing approaches for Inference in GPLVMs. Our work studies the scalable alternative with SVI across all these models.

Reference	Decoder ($X \to Y$)	Latent Variable $q(X)$	Encoder ($Y \to X$)	Training Method
Lawrence [2004]	GP	point est.	✗	Gradient based opt.
Lawrence and Quiñonero Candela [2006]	GP	point est.	✓	Gradient Based opt.
Titsias and Lawrence [2010]	GP	Gaussian	✗	Variational Inference
Bui and Turner [2015]	GP	Gaussian	✓	SVI
This work	GP	point / Gaussian	✗/✓	SVI

latent space to be automatically determined by maximisation of the ELBO. However, this ELBO does not factorise across data points [Titsias and Lawrence, 2010]. We extend the big data regression setting proposed in Hensman et al. [2013] to the unsupervised latent variable model setting. We reformulate Bayesian GPLVM for scalable inference using SVI by using a structured doubly stochastic lower bound [Salimbeni and Deisenroth, 2017]. We denote this model as *Bayesian SVI* or **B-SVI** for short.

The smooth GP decoder mapping ensures that points close in latent space are mapped to points close in data space. The notion of an *encoder* for GPLVMs was introduced in [Lawrence and Quiñonero Candela, 2006] where an additional mapping (called the *back-constraint* by the authors) was learnt expressing each latent point in the evidence (marginal likelihood) as a function of its corresponding data point. This incarnation ensured that data-space proximities were preserved in latent encodings. Hence, GPLVMs can be put on the same footing as autoencoding models with an *encoder* mapping from data to latent space and a *decoder* mapping from latent to data space. This is the third model we include in our compendium which we call *Autoencoded Bayesian SVI* or **AEB-SVI**.

In summary, our main contributions are:

- Comparison of a suite of GPLVM models which differ in the form of the latent variable but share the same inference strategy (SVI). We conduct experiments with the SVI-compatible doubly stochastic evidence lower bound for the point, maximum-a-posteriori (MAP) and Bayesian SVI models enabling efficient and scalable inference.
- Conduct experiments with a general amortised inference model which models the parameters of the Gaussian variational latent distribution using a deep neural network encoder.
- Demonstrate how training in these models is compatible with partially and massively missing data settings[1] frequently embodied in real-world datasets.

2 Background

2.1 Bayesian GPLVM

In the sparse variational formulation underlying the Bayesian GPLVM we have a training set comprising of N D-dimensional real valued observations $\mathbf{Y} \equiv \{\boldsymbol{y}_n\}_{n=1}^N \in \mathbb{R}^{N\times D}$. These data are associated with N Q-dimensional latent variables, $\mathbf{X} \equiv \{\boldsymbol{x}_n\}_{n=1}^N \in \mathbb{R}^{N\times Q}$ where $Q < D$ provides dimensionality reduction [Lawrence, 2004]. The forward mapping ($\mathbf{X} \longrightarrow \mathbf{Y}$) is governed by GPs independently defined across dimensions D. The sparse GP formulation describing the data is as follows:

$$p(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \mathbf{0}, \mathbb{I}_Q), \tag{1}$$

$$p(\mathbf{F}|\mathbf{U}, \mathbf{X}, \boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{f}_d; K_{nn}K_{mm}^{-1}\boldsymbol{u}_d, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}), \tag{2}$$

$$p(\mathbf{Y}|\mathbf{F}, X) = \prod_{n=1}^{N}\prod_{d=1}^{D} \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2), \tag{3}$$

where $\mathbf{F} \equiv \{\boldsymbol{f}_d\}_{d=1}^D$, $\mathbf{U} \equiv \{\boldsymbol{u}_d\}_{d=1}^D$ and $\boldsymbol{y_d}$ is the d^{th} column of $\mathbf{Y}$. K_{nn} is the covariance matrix corresponding to a user chosen positive-definite kernel function $k_\theta(x, x')$ evaluated on latent points $\{\boldsymbol{x}_n\}_{n=1}^N$ and parameterised by hyperparameters $\boldsymbol{\theta}$. The kernel hyperparameters are shared across all dimensions D.

[1] bulk of the dimensions missing for every data point yielding a very sparse data matrix.

The inducing variables per dimension $\{\boldsymbol{u}_d\}_{d=1}^D$ are distributed with a GP prior $\boldsymbol{u}_d|Z \sim \mathcal{N}(\mathbf{0}, K_{mm})$ computed on inducing input locations $Z \in \mathbb{R}^{M\times Q}$ which live in latent space and have dimensionality Q (matching $\boldsymbol{x}_n$).

The variational formulation,

$$p(\mathbf{F}, \mathbf{X}, \mathbf{U}|\mathbf{Y}) \approx q(\mathbf{F}, \mathbf{X}, \mathbf{U}) = \Big[\prod_{d=1}^{D} p(\boldsymbol{f}_d|\boldsymbol{u}_d, X)q(\boldsymbol{u}_d)\Big]q(\mathbf{X}) \tag{4}$$

admits a tractable lower bound to the marginal likelihood $p(\mathbf{Y}|\boldsymbol{\theta})$ where the inducing variables are integrated out or *collapsed* [Titsias and Lawrence, 2010].

The original bound incorporated the optimal Gaussian variational distribution $q^*(\boldsymbol{u}_d)$ and a diagonal Gaussian variational distribution, $q(\mathbf{X}) = \prod_{n=1}^N \mathcal{N}(\boldsymbol{x}_n; \mu_n, s_n\mathbb{I}_Q)$,. However, every gradient step needs a pass over the full dataset of size N. In the section below we describe the Bayesian SVI model which uses the same variational formulation as above except for the treatment of the inducing variables per dimension $\boldsymbol{u}_d$. Instead of using their optimal analytic form, we learn their parameters through direct optimisation of the *uncollapsed* lower bound.

3 Generalised GPLVM with SVI

For the SVI bound we keep the representation of $\mathbf{U}$ uncollapsed; we learn a mean and dense covariance matrix numerically using stochastic gradient methods on $q(\boldsymbol{u}_d) \sim \mathcal{N}(\boldsymbol{m}_d, S_d)$.

3.1 Is SVI applicable?

Stochastic Variational Inference (SVI) [Hoffman et al., 2013] pre-requisites a joint probability model with a set of global and local hidden variables where the local variables are conditionally independent given the global variables. GP models for regression in their standard form do not admit such a factorisation and neither do they have global variables, however Hensman et al. [2013] showed how the SVI machinery becomes applicable by introducing global inducing variables $\boldsymbol{u}$ and variationally marginalising $\boldsymbol{f}$. We assume a single output dimension in this sub-section hence drop the dimension index d.

$$\ln p(\boldsymbol{y}|\boldsymbol{u}) = \ln \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{u})d\boldsymbol{f} \geq \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})}[\ln p(\boldsymbol{y}|\boldsymbol{f})] \triangleq \ln \tilde{p}(\boldsymbol{y}|\boldsymbol{u}) \tag{5}$$

where $\tilde{p}(\boldsymbol{y}|\boldsymbol{u})$ factorises if the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$ does.

$$\tilde{p}(\boldsymbol{y}|\boldsymbol{u}) = \prod_{n=1}^{N} \mathcal{N}(y_n|k_n^T K_{mm}^{-1}\boldsymbol{u}, \sigma_n^2) \exp\left\{-\frac{1}{2\sigma_y^2}(k_{nn} - k_n^T K_{mm}^{-1} k_n)\right\} \tag{6}$$

where k_n is the n^{th} column of K_{mn} (only dependent on point $\boldsymbol{x}_n$).

We now have a model with global variables and a likelihood which is conditionally independent across observations given the global variables $\boldsymbol{u}$. The regression model does not need local hidden variables. However, in the latent variable setting we have a latent variable $\boldsymbol{x}_n$ per training point (see supplementary for graphical models.)

3.2 Doubly Stochastic Evidence Lower bound (DS-ELBO)

Doubly stochastic was proposed by Titsias and Lázaro-Gredilla [2014] and deployed in deep Gaussian process regression by Salimbeni and Deisenroth [2017]. Here we use doubly stochastic inference in the unsupervised latent variable setting, where the aim is dimensionality reduction.

Keeping with the formulation in section 2.1 we write down the rudimentary ELBO,

$$\mathcal{L} = p(\mathbf{F}|\mathbf{U}, \mathbf{X})q(\mathbf{U})q(\mathbf{X}) \log \frac{p(\mathbf{Y}|\mathbf{F}, \mathbf{X})p(\mathbf{U}|Z)p(\mathbf{X})}{q(\mathbf{U})q(\mathbf{X})} d\mathbf{F}d\mathbf{U}d\mathbf{X} \tag{7}$$

Making the parameterisation of the variational distributions explicit for clarity, we denote the variational distribution over the latent points as $q_\phi(\boldsymbol{x}_n)$ where $\phi = \{\mu_n, s_n\}$ and the variational

distribution over the inducing variables as $q_\lambda(\boldsymbol{u}_d)$ where $\lambda = \{\boldsymbol{m}_d, S_d\}$. We re-write equation 7 with the familiar decomposition involving the expected log-likelihood term and KL terms,

$$\begin{aligned}\mathcal{L}(\mathcal{D}) &= \mathbb{E}_{q(.)}[\log p(\mathbf{Y}|\mathbf{F},\mathbf{X})] - \mathrm{KL}(q(\mathbf{X})||p(\mathbf{X})) - \mathrm{KL}(q(\mathbf{U})||p(\mathbf{U})) \qquad (8)\\ &= \mathbb{E}_{q(.)}[\sum_{n,d}\log\mathcal{N}(y_{n,d};\boldsymbol{f}_d(\boldsymbol{x}_n),\sigma_y^2)] - \sum_n \mathrm{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)) - \sum_d \mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))\\ &= \sum_{n,d}\mathbb{E}_{q_\phi(\boldsymbol{x}_n)}[\underbrace{\mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)q_\lambda(\boldsymbol{u}_d)}[\log\mathcal{N}(y_{n,d};\boldsymbol{f}_d(\boldsymbol{x}_n),\sigma_y^2)]}_{\mathcal{L}_{n,d}(\boldsymbol{x}_n,y_{n,d})=\mathcal{L}_{n,d}}] - \sum_n \mathrm{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)) \qquad (9)\\ &\quad - \sum_d \mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))\end{aligned}$$

The expected log-likelihood term for a single data point (n) and dimension (d) - $\mathcal{L}_{n,d}(\boldsymbol{x}_n, y_{n,d})$ is reduced to,

$$\begin{aligned}\mathbb{E}_{q_\phi(\boldsymbol{x}_n)}[\mathcal{L}_{n,d}] &= \int q_\phi(\boldsymbol{x}_n)\left[\int q_\lambda(\boldsymbol{u}_d)\left[\int p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)\log\mathcal{N}(y_{n,d};\boldsymbol{f}_d(\boldsymbol{x}_n),\sigma_y^2)d\boldsymbol{f}_d(\boldsymbol{x}_n)\right]d\boldsymbol{u}_d\right]d\boldsymbol{x}_n\\ &= \log\mathcal{N}(y_{n,d}|\underbrace{\langle k(\boldsymbol{x}_n,Z)\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_1^{(n,\cdot)}}K_{mm}^{-1}\boldsymbol{m}_d,\sigma_y^2) - \frac{1}{2\sigma_y^2}\mathrm{Tr}(\underbrace{\langle k(\boldsymbol{x}_n,\boldsymbol{x}_n)\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\psi_0^n}) \qquad (10)\\ &\quad + \frac{1}{2\sigma_y^2}\mathrm{Tr}(K_{mm}^{-1}\underbrace{\langle k(Z,\boldsymbol{x}_n)k(\boldsymbol{x}_n,Z)\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_2^n}) - \frac{1}{2\sigma_y^2}\mathrm{Tr}(S_dK_{mm}^{-1}\underbrace{\langle k(Z,\boldsymbol{x}_n)k(\boldsymbol{x}_n,Z)\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_2^n}K_{mm}^{-1})\end{aligned}$$

where we analytically perform the integration w.r.t $q_\lambda(\boldsymbol{u}_d)$ and the inner-most integral w.r.t $p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)$ is the same as eq. 6 leaving behind the expectations w.r.t $q_\phi(\boldsymbol{x}_n)$ which are handled numerically with Monte Carlo estimation.

$$\Psi^{(n,\cdot)} \approx \frac{1}{J}\sum_{j=1}^J k(\boldsymbol{x}_n^{(j)},Z), \quad \Psi_2^n \approx \frac{1}{J}\sum_{j=1}^J k(Z,\boldsymbol{x}_n^{(j)})k(\boldsymbol{x}_n^{(j)},Z), \quad \psi_0^n \approx \frac{1}{J}\sum_{j=1}^J k(\boldsymbol{x}_n^{(j)},\boldsymbol{x}_n^{(j)}) \qquad (11)$$

where $\boldsymbol{x}_n^{(j)} \sim q_\phi(\boldsymbol{x}_n)$; the samples $\boldsymbol{x}_j$ are drawn using the reparameterization trick Kingma and Welling [2014] where we sample $\epsilon^{(j)} \sim \mathcal{N}(0,\mathbb{I}_Q)$ and $\boldsymbol{x}_n^{(j)} = \mu_n + s_n \odot \epsilon^{(j)}$.

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_n)}[\mathcal{L}_{n,d}] \simeq \frac{1}{J}\sum_{j=1}^J \mathcal{L}_{n,d}(\boldsymbol{x}_n^{(j)},y_{n,d}) = \frac{1}{J}\sum_{j=1}^J \mathcal{L}_{n,d}(\mu_n+s_n\odot\epsilon^{(j)},y_{n,d}) = \frac{1}{J}\sum_{j=1}^J \mathcal{L}_{n,d}(g_\phi(\epsilon^{(j)}),y_{n,d})$$

We denote the approximate ELBO as $\hat{\mathcal{L}}(\mathcal{D})$,

$$\hat{\mathcal{L}}(\mathcal{D}) = \sum_n\sum_d \overbrace{\frac{1}{J}\sum_{j=1}^J \mathcal{L}_{n,d}(g_\phi(\epsilon^{(j)}),y_{n,d})}^{\hat{\mathcal{L}}_{n,d}} - \sum_d \mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z)) - \sum_n \mathrm{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)) \qquad (12)$$

The sparse GPLVM model in experiments comprises of just the first two terms in eq. 12, while the MAP method excludes the KL divergence term for latents $(\boldsymbol{x}_n)$ in exchange for solely the prior term $p(\boldsymbol{x}_n)$. Finally, in order to speed-up computation we use mini-batches, where in each gradient step we take a random sample $B < N$ of the data-points $\mathcal{D}_B \equiv \{y_b\}_{b=1}^B, \mathcal{D}_B \subset \mathcal{D}$ to construct a scalable, differentiable and unbiased estimator optimised with standard stochastic gradient methods. The KL terms are analytically tractable due to the choice of the Gaussian variational family for $q_\phi(\boldsymbol{x}_n)$ and the optimal (Gaussian) variational family for $q_\lambda(\boldsymbol{u}_d)$.

The method is known as *doubly stochastic variational inference* due to the two-fold stochasticity attributed to computing the expectations with Monte Carlo and due to mini-batching.

Algorithm 1: Bayesian GPLVM with Doubly Stochastic Variational Inference (**B-SVI**)

Input: ELBO objective $\mathcal{L}$, gradient based optimiser `optim()`, training data $\mathcal{D} = \{\boldsymbol{y}_n\}_{i=1}^N$
Initial model params:
$\boldsymbol{\theta}$ (covariance hyperparameters for GP mappings $\boldsymbol{f}_d$ and data noise variance σ_y^2),
Initial variational params:
$Z \in \mathbb{R}^{M \times Q}$ (inducing locations),
$\phi = \{\mu_n, s_n\}_{n=1}^N$ (local variational parameters - $\boldsymbol{x}_n \sim \mathcal{N}(\mu_n, s_n \mathbb{I}_Q), \mu_n, s_n \in \mathbb{R}^Q$)
$\lambda = \{m_d, S_d\}_{d=1}^D$ (global variational parameters - $\boldsymbol{u}_d \sim \mathcal{N}(m_d, S_d), \boldsymbol{u}_d \in \mathbb{R}^M, S_d \in \mathbb{R}^{M \times M}$)

while *not converged* **do**

- Choose a random mini-batch $\mathcal{D}_B \subset \mathcal{D}$.
- Sample J samples from the noise distribution $\epsilon^{(j)} \sim \mathcal{N}(0, \mathbb{I}_Q)$.
- Form a mini-batch estimate of the ELBO:
$$\hat{\mathcal{L}}(\mathcal{D}_B) = \frac{N}{B}\left(\sum_b \sum_d \hat{\mathcal{L}}_{b,d} - \sum_b \mathrm{KL}(q_\phi(\boldsymbol{x}_b)||p(\boldsymbol{x}_b)\right) - \sum_d \mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))$$
- Gradient step: $Z, \boldsymbol{\theta}, \sigma_y^2, \{\mu_b, s_b\}_{b=1}^B \{m_d, S_d\}_{d=1}^D \longleftarrow \texttt{optim}(\hat{\mathcal{L}}(\mathcal{D}_B))$

end
return $Z, \boldsymbol{\theta}, \phi, \lambda$

3.3 Amortised Inference with Encoders

The GPLVM model provides a probabilistic non-linear mapping from latent space $\mathbf{X}$ to data space $\mathbf{Y}$. A probabilistic representation in the latent space provides several advantages - 1) a representation of uncertainty in the latent encoding can be valuable for downstream tasks and 2) we can sample points from the latent space to reconstruct data in the observation space by passing them through the trained decoder. However, the GPLVM inherently preserves dissimilarities in data space. It ensures that two points which are apart in data space, also apart in latent space. This is due to the smooth GP mapping from $\mathbf{X} \rightarrow \mathbf{Y}$. Local distances are preserved in the latent space ensuring that points *close*[2] in latent space recover observations that are close in data space. Lawrence and Quiñonero Candela [2006] and Bui and Turner [2015] additionally account for this feature of data distance preservation by introducing an encoder within the GPLVM model (see also [Dai et al., 2016]). Parameters of the variational distribution of each latent point $\boldsymbol{x}_n$ are reparameterised as a function of the data in the objective, thereby introducing a dependency for each latent point on its corresponding data point. This function is usually referred to as the back-constraint and its parameters are *global*, i.e. shared between all the data points. This allows for fast amortised inference and constant time test predictions.

AEB-SVI: In this variational model, the mean and variance of the base Gaussian distribution are parameterised as outputs of individual neural networks G_{ϕ_1} and H_{ϕ_2} with network weights ϕ_1 and ϕ_2. The network weights are shared across all the data points enabling amortised learning [Bui and Turner, 2015]. The key property of this parameterisation is that it learns a dense covariance matrix (through Cholesky decomposition) per data-point thereby capturing correlations across dimensions in latent space.

$$q(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; G_{\phi_1}(\boldsymbol{y}_n), H_{\phi_2}(\boldsymbol{y_n})^T H_{\phi_2}(\boldsymbol{y_n})) \tag{13}$$

3.4 Predictions

When unseen high-dimensional points arrive in data space $\boldsymbol{y}^*$ we are interested in computing the latent point distribution $q(\boldsymbol{x}^*)$ per test point $\boldsymbol{y}^*$ where we have access to the trained variational parameters (ϕ, Z, λ) and model hyperparameters $(\boldsymbol{\theta})$. One motivation for auto-encoder driven models is that we have constant-time $\mathcal{O}(1)$ test predictions. Given a test point $\boldsymbol{y}^*$, we use the set of global encoder weights (ϕ_1, ϕ_2) to obtain the posterior approximation $q(\boldsymbol{x}^*)$ (as in eq. 13). In the Bayesian SVI model (Algorithm 1.) we can't obtain the distributional parameters for $q(\boldsymbol{x}^*)$ deterministically, instead we re-optimise the ELBO with the additional test data point $\boldsymbol{y}^*$ while keeping all the global and

[2]For a stationary kernel, this would be closeness in a sense of Euclidean distance.

model hyperparameters frozen at their trained values. Note that since the SVI ELBO factorises across data points, $\mathcal{L}(\{\boldsymbol{y}_n\}_{n=1}^N, \boldsymbol{y}^*) = \sum_{n=1}^{N+1}\sum_{d=1}^{D}\mathcal{L}_{n,d}$, the gradients to derive the distributional parameters of the test point $\mathcal{N}(\mu_*, s_*\mathbb{I}_Q)$ only depend on the component terms.

3.5 Training with missing dimensions

A key motivation for our framework is dealing with missing data at *training time*. Most machine learning algorithms are designed to be deployed on carefully curated tables of data with a fixed number of features. If data is missing, it is often dealt with through EM algorithms which can deal with missingness up to around 30%. In the real world the situation is often very different. Important data sets such as electronic health records can have 90% or more missing values. In these domains the objective function becomes dominated by the missing values and learning fails to occur [Corduneanu and Jaakkola, 2002]. We consider a data set-up where every vector $\boldsymbol{y}$ has an arbitrary number of dimensions missing and there is no constraint or structure about their *missingness*. Our training procedure leverages the marginalisation principle of Gaussian distributions and the fact that the data dependent terms of the SVI ELBO factorise across data points and dimensions. This means we can trivially marginalise out the missing dimensions $\boldsymbol{y}_a$, because each individual data point $\boldsymbol{y}$ is modelled as a joint Gaussian. Consider a high-dimensional point $\boldsymbol{y}$ which we split into observed, $\boldsymbol{y}_o$ and unobserved $\boldsymbol{y}_a$ dimensions,

$$\int \prod_{d\in a}\prod_{d\in o} p(\boldsymbol{y}_a, \boldsymbol{y}_o|\boldsymbol{u}_d, \mathbf{X})d\boldsymbol{y}_a = \prod_{d\in o} p(\boldsymbol{y}_o|\boldsymbol{u}_d, \mathbf{X}), \tag{14}$$

where a and o denote the indices of missing and observed dimensions respectively and all dimensions are given as, $D = a \cup o$. $\boldsymbol{u}_d \in \mathbb{R}^M$ denote the inducing variables which ensure conditional independence. The latent variables per data point $\boldsymbol{x}_n$ are informed by the observed dimensions only, while the M inducing variables per dimension $\boldsymbol{u}_d s$ are informed by all the data points which have the observed dimension. The elegance of this framework is that there is no major change in the training procedure as the ELBO eq. 12 sums over all observed dimensions per data point. We can also easily reconstruct the missing training dimensions by decoding the mean of the optimised variational latent distribution $q(\boldsymbol{x}) = \mathcal{N}(\mu^*, s^*\mathbb{I}_Q)$.

$$p(\boldsymbol{y}_{a\cup o}|\boldsymbol{y}_o) = \prod_{d=1}^{D}\mathcal{N}(\boldsymbol{y}_d; \boldsymbol{f}_d(\mu^*), \sigma_y^2), \tag{15}$$

where $\boldsymbol{f}_d$ is a draw from the sparse GP prior eq. 2 with covariance matrices computed with optimised model and variational parameters. This set-up reflects real-world data which is often sparse with many missing and few overlapping dimensions across the full dataset. The experiments in section 4.2 demonstrate the reconstruction ability of Bayesian SVI when faced with missing dimensions at training time. The missing data framework is not immediately compatible with auto-encoding models as every latent point $\boldsymbol{x}_n$ is expressed as a function of the data point $\boldsymbol{y}_n$. However, set encoders [e.g Qi et al., 2017; Vedantam et al., 2017; Ma et al., 2018] can also be seamlessly integrated as the auto-encoding component with the GPLVM. We defer this to future work.

4 Experiments

4.1 Ablation Study: Benchmarks

Models: Our experiments implement four incarnations of the GPLVM model namely, POINT which refers to the Sparse GPLVM, MAP which refers to the sparse GPLVM with a prior over latent variables $\boldsymbol{x}_n$, the Bayesian SVI model B-SVI and AEB-SVI which refers to the Autoencoded Bayesian GPLVM. For each model we record the training and test reconstruction error (RMSE) and final ELBO loss. Full details about the experimental set-up are enclosed in the supplementary material.

Data set-up: The multi-phase Oilflow data [Bishop and James, 1993] consists of 1000, $12d$ data points belonging to three classes which correspond to the different phases of oil flow in a pipeline. The qPCR data contains 48 dimensional single-cell data obtained from mice [Guo et al., 2010] where

each dimension corresponds to a gene. Cells differentiate during their development and these data were obtained at various stages of development which contribute 10 categories/classes to which each of the cell belongs. We use a 80/20 split for training/testing and report test performance with $\pm$ 2 standard errors over three optimization runs.

Table 2: Test RMSE for datasets with $\pm$ standard error across 3 optimisation runs. Z denotes the number of inducing variables used per dimension and Q denotes the dimension of the latent space.

Dataset	N / d	Z	Q	POINT	MAP	B-SVI	AEB-SVI
Oilflow	1000 / 12	25	10	0.341 (0.008)	0.569 (0.092)	0.0925 (0.025)	**0.067 (0.0016)**
qPCR	450 / 48	40	11	0.624 (0.027)	0.589 (0.016)	**0.554 (0.017)**	**0.539 (0.004)**

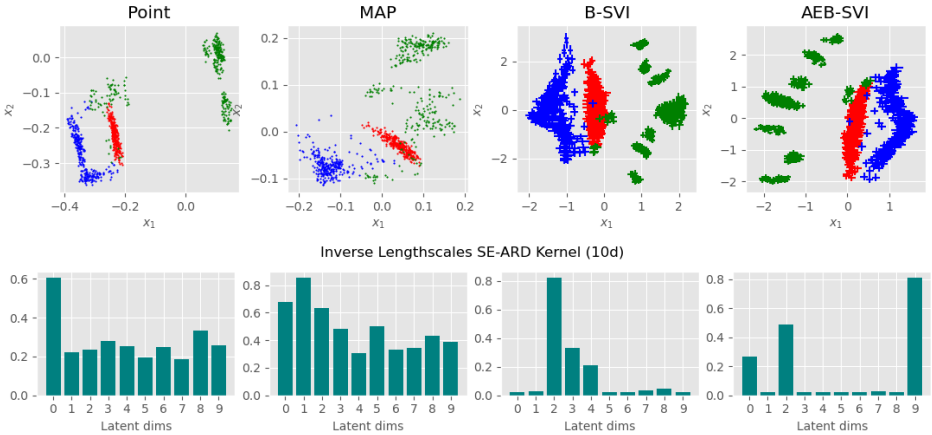

Figure 1: Top: The 2d latent space corresponding to the dominant dimensions learnt by each model. Bottom: The inverse lengthscales learnt by each model specification. We include a similar report for qPCR in the supplementary.

We trained each of the 4 models on two high-dimensional datasets and summarise results in table 2. The $2d$ projections of the latent space clearly show that all variants are able to discover the class structure. It is important to note that unlike previous versions these models do not require PCA initialisation and all models were initialised randomly. In order to highlight certain features, the latent dimensionality (Q) was kept fixed across all models.

POINT and MAP overfit as can be seen from the magnitude of the inverse lengthscales across all the latent dimensions. Both POINT and MAP find all the latent dimensions relevant. Conversely, B-SVI and AEB-SVI identify two or three dominant dimensions to represent the data.

The training/test error comparison 2 provides further evidence of overfitting in the point methods. The quality of the 2d latent projection of training data using the fully trained model might hide the overfitting effects as it is extremely effective at disentangling class structure in training data. However, it is important to look beyond the quality of the 2d projection before passing them to downstream tasks.

We show additional analysis in the supplementary where the Bayesian methods with SVI don't overfit even when we match the latent space dimensionality to that of the data space. This analysis underscores the importance of the KL term over latents in ELBO objective. Mathematically, the inclusion or exclusion of this term is the main fundamental difference in these formulations. It is further interesting to note that MAP underperforms Point in both examples and the presence of solely the prior term in the SVI ELBO leads to worse performance than canonical optimisation for point estimates.

4.2 Missing data: Reconstructing structured images

The focus of this experiment is to qualitatively assess how the models capture uncertainty when training with missing data in structured inputs. We use 70000 training samples from the MNIST digits dataset [LeCun et al., 2010] with $\approx 60\%$ of the pixels missing at random in each digit. Each

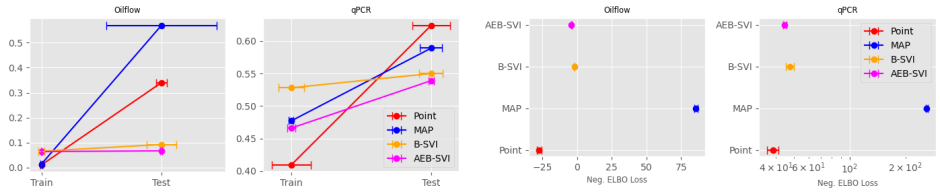

Figure 2: Left: The train and test RMSE per model. Right: The final converged negative ELBO loss. Note that the ELBO loss formulation is different across the point models and the only identical comparison is between the B-SVI and AEB-SVI models.

image has 768 pixels yielding a $768d$ data space. The image data set [Roweis and Saul, 2000] contains $\approx$2000 images of a face taken from sequential frames of a short video. Each image is of size 20x28 yielding a $560d$ data space. Fig. 3 summarises sample generation from the learnt 2d latent distribution. Note that this reconstruction experiment differs from the less challenging *test-time* missing data which has been demonstrated in several works Titsias and Lawrence [2010]; Gal et al. [2014]. We include results for test-time missing reconstruction in the supplementary while focussing on the training-time missing data scenario here.

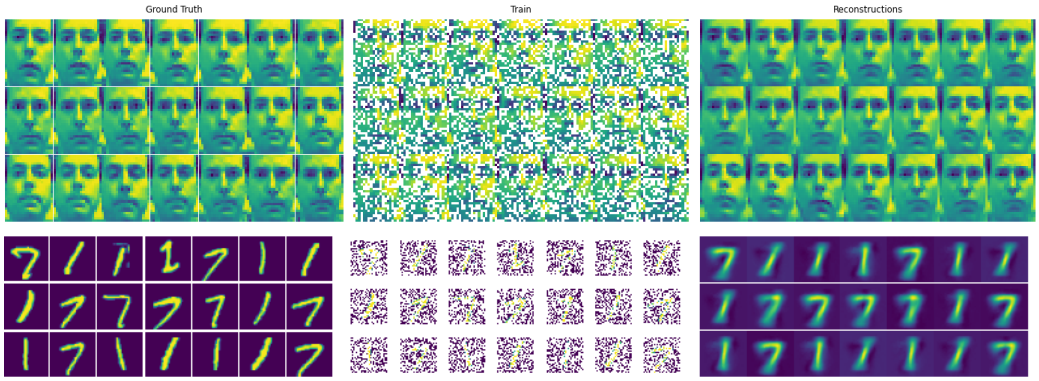

Figure 3: Top Row: Brendan faces reconstruction task with 39% missing pixels. Bottom row: MNIST reconstruction task where the digits were trained on partially observed images. In both rows the left column denotes ground truth data, the center column denotes a subset of the training data and the right column denotes reconstructions from a 2d and 5d latent distribution for MNIST and Brendan respectively. We include more examples in supplementary material.

To demonstrate the versatility of the reconstruction task we tested the method on several examples of the *walking* human pose from the CMU motion capture database. We split up these motions into four sections, and remove an assortment of body components. We then try to recreate the entire body movement using B-SVI using the learnt latent means. A sample reconstruction is shown in fig. 4.

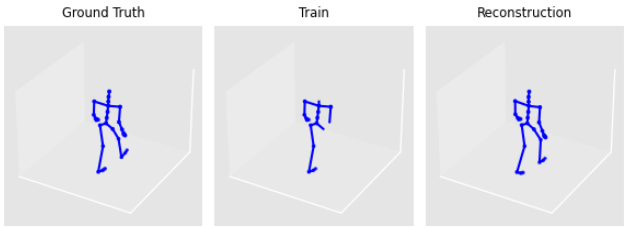

Figure 4: The reconstruction of a single high-dimensional human pose.

4.3 Massively Missing data

In this section we demonstrate how the B-SVI model can be used to train with *massively* missing data. We define the massively missing data to be when greater than 90% of the values are missing. We train in a purely unsupervised way and demonstrate state-of-the-art test-time prediction performance. Note that we make no use of ad-hoc zero imputation or meta-features to boost prediction performance. We

rely on the Gaussian process formalism for hyperparameter tuning, avoiding the need for expensive cross-validation and tuning which is required for many deep NN architectures. The algorithms only parameters are the batch size and learning rate. They are set to 100 and 0.001 respectively with the ADAM optimizer [Kingma and Ba, 2015].

4.3.1 MovieLens100K

The movie lens 100K data has 1682 movies (columns/D) across 943 users (rows/N) where each user has rated an average of 20 movies [Harper and Konstan, 2015]. The ratings range from $\{1, 2, \ldots, 5\}$. This yields an extremely sparse data grid with 93.8% of the entries missing.[3] We learn a $10d$ latent distribution for the movie lens data and summarise the test results in table 3. We trained on 90% users and made predictions for 10% of the users. The test performance captures the ability of the model to predict a rating for a new user on any of the movies in the database. We reconstruct test ratings for users and report performance below with baselines from literature.

Table 3: Test RMSE score of B-SVI compared to matrix factorisation method.

Dataset / Method	PMF	BiasMF	NNMF	B-SVI
MovieLens100K	0.952	0.911	0.903	0.924

PMF scores were taken from Mnih and Salakhutdinov [2007] and the BiasMF / NNMF scores from Dziugaite and Roy [2015].

5 Related Work

GPLVM & Variants: The GPLVM model has spawned several variants since its introduction in Lawrence [2004]. The most fundamental variants are summarised in table 1. Apart from these there has been a suite of work extending the canonical Bayesian GPLVM model to target different objectives. [Damianou et al., 2016] provides a rigorous examination of the evidence lower bound in the Bayesian GPLVM formulation and extends it to multiple scenarios which include high-dimensional time-series [Damianou et al., 2011] and uncertain inputs for GP regression. The shared GPLVM model [Ek et al., 2007] considers a generative model with multiple sources of data and learns a shared representation in the latent space, capable of generating data in the joint observation space. [Gal et al., 2014] reformulate the Bayesian GPLVM enabling a distributed inference algorithm. Urtasun and Darrell [2007] use GPLVMs in the context of classification using discriminative priors in latent space and Urtasun et al. [2008] focus on embedding data in non-Euclidean latent spaces which is useful when high-dimensional data lie on a natural manifold, e.g. human motion. Other relevant works include [Dai et al., 2016] which augment a deep GP with a recognition model for latent variable inference. None of these works use SVI for inference in these models.

Other related work: In terms of applications, the GPLVM has been widely used in the biological sciences [Ahmed et al., 2019], [Verma and Engelhardt, 2020] and engineering domains, with the most prominent applications in microarray qPCR datasets to infer the evolution of branching structure in genes [Campbell and Yau, 2015].

6 Conclusion

This paper introduces a generalised inference strategy for GPLVM models with key properties like parallelisable inference, auto-encoding for fast test time inference, and the ability to handle missing data during training. The non-parametric nature of the Gaussian process decoder makes this framework unique to deep parameteric latent variable models like VAEs. We showed in experiments that a fully Bayesian training procedure in conjunction with SVI yields state-of-the-art test time performance. Amortisation with a deep NN encoder introduces local correlations between dimensions in latent space without hampering test performance. A key characteristic of our model is its ability to train in the *massively missing data* regime that is inadequately addressed by modern parametric machine learning models. The approach can be extended to learn richer variational families in latent space along with missing data. Future work would focus in that direction.

[3] each row denotes a user, when a user has not rated a movie the value is NaN.

References

S. Ahmed, M. Rattray, and A. Boukouvalas. GrandPrix: scaling up the Bayesian GPLVM for single-cell data. *Bioinformatics*, 35(1):47–54, 2019.

C. M. Bishop and G. D. James. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 327(2-3):580–593, 1993.

T. D. Bui and R. E. Turner. Stochastic variational inference for Gaussian process latent variable models using back constraints. In *Black Box Learning and Inference NIPS workshop*, 2015.

K. Campbell and C. Yau. Bayesian Gaussian process latent variable models for pseudotime inference in single-cell rna-seq data. *bioRxiv*, page 026872, 2015.

A. Corduneanu and T. Jaakkola. Continuation methods for mixing heterogeneous sources. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, page 111–118, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608974.

Z. Dai, A. C. Damianou, J. González, and N. D. Lawrence. Variational auto-encoded deep gaussian processes. In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1511.06455.

A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2011.

A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational inference for latent variables and uncertain inputs in Gaussian processes. *The Journal of Machine Learning Research*, 17(1): 1425–1486, 2016.

G. K. Dziugaite and D. M. Roy. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*, 2015.

C. H. Ek, P. H. S. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *International workshop on machine learning for multimodal interaction*, pages 132–143. Springer, 2007.

Y. Gal, M. Van Der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *Advances in Neural Information Processing Systems*, pages 3257–3265, 2014.

G. Guo, M. Huss, G. Q. Tong, C. Wang, L. L. Sun, N. D. Clarke, and P. Robson. Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Developmental cell*, 18(4):675–685, 2010.

F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM transactions on interactive intelligent systems (TIIS)*, 5(4):1–19, 2015.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, 2013.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(4):1303–1347, 2013. URL http://jmlr.org/papers/v14/hoffman13a.html.

I. T. Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, volume 3, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

N. D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GPLVM through back constraints. In *Proceedings of the 23rd international conference on Machine learning*, pages 513–520, 2006.

Y. LeCun, C. Cortes, and C. J. Burges. MNIST handwritten digit database. 2010. *URL http://yann. lecun. com/exdb/mnist*, 7:23, 2010.

C. Ma, S. Tschiatschek, K. Palla, J. M. Hernández-Lobato, S. Nowozin, and C. Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018.

A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2007.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes in machine learning*. Springer, 2006.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

H. Salimbeni and M. Deisenroth. Doubly stochastic variational inference for deep gaussian processes. *arXiv preprint arXiv:1705.08933*, 2017.

M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

M. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

M. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979. PMLR, 2014.

R. Urtasun and T. Darrell. Discriminative Gaussian process latent variable model for classification. In *Proceedings of the 24th international conference on Machine learning*, pages 927–934, 2007.

R. Urtasun, D. J. Fleet, A. Geiger, J. Popović, T. J. Darrell, and N. D. Lawrence. Topologically-constrained latent variable models. In *Proceedings of the 25th international conference on Machine learning*, pages 1080–1087, 2008.

R. Vedantam, I. Fischer, J. Huang, and K. Murphy. Generative models of visually grounded imagination. *arXiv preprint arXiv:1705.10762*, 2017.

A. Verma and B. E. Engelhardt. A robust nonlinear low-dimensional manifold for single cell rna-seq data. *BMC bioinformatics*, 21(1):1–15, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] This paper analyses SVI inference across different variants of a model class. This is best summarised in the introduction.
   (b) Did you describe the limitations of your work? [Yes] We do include a section dedicated to this in the supplementary.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Since this work is on methodology we did not deem it necessary to discuss societal impacts.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We concur.

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We do include a mathematical derivation with assumptions.
   (b) Did you include complete proofs of all theoretical results? [Yes] Section 3.2
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] For full details please refer to the experimental configuration section in the supplementary
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All relevant attributes which could impact reproduction is included both in the experiment sections and supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to the experimental configuration section in the supplementary
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We do cite all the original datasets and the gpytorch library used for experiments.
   (b) Did you mention the license of the assets? [Yes] We will include this where necessary in the supplementary.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We include the license of assets in the main paper in the supplementary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We used publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]