3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–12

# Gaussian Process Latent Variable Flows for Massively Missing Data

**Vidhi Lalchand** VR308@CAM.AC.UK
*University of Cambridge, UK*

**Aditya Ravuri** ADITYA.RAVURI@CANTAB.NET

**Neil D. Lawrence** NDL21@CAM.AC.UK
*University of Cambridge, UK*

## Abstract

Gaussian process latent variable models (GPLVM) are used to perform nonlinear and probabilistic dimensionality reduction. They extend Gaussian processes (GP) to the domain of unsupervised learning (Lawrence, 2004). The Bayesian incarnation of the GPLVM uses a variational framework, where the posterior over all unknown quantities is approximated by a well-behaved variational family, a factorised Gaussian (Titsias and Lawrence, 2010). This gives not only implicit regularisation but also mathematical convenience. In this work we narrow our focus on examining the quality of the latent representation learnt under this Gaussian assumption. We introduce non-Gaussianity in the distribution of the latent space through *normalising flows*. The flexibility afforded by flows is critical in modelling massively missing data. Inference is performed using Stochastic Variational Inference (SVI) with a structured variational lower bound that factorizes across data points permitting efficient and scalable mini-batching of gradients. We call this flexible model class *Gaussian process latent variable flows* (GPLVF). We compare this framework with traditional models like the Bayesian GPLVM. Our experiments focus on massively missing data settings.

## 1. Introduction

Gaussian processes (GPs) represent a powerful non-parametric probabilistic framework for performing regression and classification with inductive biases 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. The GPLVM provides a probabilistic mapping from (an unobserved) latent space to data-space where a GP acts as a *decoder*, with the smoothness of the mapping controlled by a kernel function. Several traditional dimensionality reduction models learn a projection of high dimensional data to lower dimensional manifolds (the direction of the mapping is reversed in a GPLVM). An important attribute of the smooth GP decoder mapping in the GPLVM is it 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.

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 relying on inducing variables (Titsias, 2009) serves a dual purpose of making the lower

Table 1: Existing approaches for Inference in GPLVMs.

| Reference | Decoder ($X \rightarrow Y$) | Latent Variable $q(X)$ | Encoder ($Y \rightarrow X$) | 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 | Flexible | ✓ | SVI |

bound to the marginal likelihood tractable, and providing computational savings. The Bayesian GPLVM also allows for the dimensionality of the latent space to be automatically determined by using ARD (*automatic relevance determination*) kernels. It prunes dimensions which correspond to a small inverse lengthscale.

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 inducing variables. The key idea was re-formulating the evidence lower bound (ELBO) in Titsias and Lawrence (2010) in a way that factorizes across the data enabling mini-batching for gradients. However, in Hensman et al. (2013) this was only explored for regression (with a Gaussian likelihood) settings rather than the unsupervised latent variable model setting.

This work looks at the Bayesian GPLVM through the lens of SVI. The generalised SVI scheme allows for richer non-Gaussian variational families through the use of normalising flows (Rezende and Mohamed, 2015). The model introduced in this work allows the distribution around the latent encoding to be flexible and expressive by warping the Gaussian variational distribution through a sequence of invertible transformations. The departure from Gaussian uncertainty means we can model latent variables as being driven by multi-modal/complex distributions which may provide a more faithful approximation to the true unknown posterior in several settings. Concretely, we summarise the main contributions of this work below:

- We extend the Bayesian GPLVM framework to non-Gaussian variational distributions through normalising flows.
- We demonstrate how this class of models can be used on datasets which are extremely sparse (bulk of the features are missing for every data point) - we call this framework *massively missing data* which is frequently embodied in real-world datasets.

## 2. Review of Bayesian GPLVM

In this section we provide an overview of the Bayesian GPLVM (Titsias and Lawrence, 2010). We have a training set comprising of $N$ $D$-dimensional real valued observations $Y \equiv \{\boldsymbol{y}_n\}_{n=1}^{N} \in \mathbb{R}^{N \times D}$. These data are associated with $N$ $Q$-dimensional latent variables, $X \equiv \{\boldsymbol{x}_n\}_{n=1}^{N} \in \mathbb{R}^{N \times Q}$ where $Q < D$ provides dimensionality reduction (Lawrence, 2004). The forward mapping ($X \longrightarrow Y$) is governed by GPs independently defined across dimensions $D$. The probabilistic model describing the data is as follows:

$$\text{Prior on latents: } p(X) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{0}, \mathbb{I}_Q),$$

$$\text{Prior on mapping: } p(\boldsymbol{f}|X, \boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{f}_d; 0, K_{nn}^{(d)}),$$

$$\text{Data likelihood: } p(Y|\boldsymbol{f}, X) = \prod_{n=1}^{N} \prod_{d=1}^{D} \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2),$$

where $\boldsymbol{f} \equiv \{\boldsymbol{f}_d\}_{d=1}^{D}$ and $\boldsymbol{y_d}$ is the $d^{th}$ column of $Y$. $K_{nn}^{(d)}$ is the covariance matrix corresponding to a user chosen positive-definite kernel function $k_\theta(x, x^{'})$ evaluated on latent points $\{\boldsymbol{x}_n\}_{n=1}^{N}$ parameterised by some hyperparameters $\boldsymbol{\theta}$. Further, $\boldsymbol{f}_d$ is a draw from a GP with covariance matrix $K_{nn}^{(d)}$. If same kernel function $k_\theta(x, x')$ is chosen across the dimensions, then $K_{nn}^{(d)}$ is identical across dimensions $d$ and we can drop the superscript.
Exact Bayesian inference in this set-up entails finding the marginal likelihood obtained by integrating out the mapping $\boldsymbol{f}$ and latent variables $X$; unfortunately this is intractable owing to the integration over the latent variables $X$[1].

$$p(Y|\boldsymbol{\theta}) = \int \dots \int p(Y|\boldsymbol{f}, X)p(\boldsymbol{f}|X, \boldsymbol{\theta})p(X)d\boldsymbol{f}_1 \dots d\boldsymbol{f}_D dX = \int \prod_{d=1}^{D} p(\boldsymbol{y}_d|X)p(X)dX, \tag{1}$$

where $p(\boldsymbol{y}_d|X) = \mathcal{N}(\mathbf{0}, K_d + \sigma_y^2 \mathbb{I})$

Further, the posterior over all the unknown quantities $p(\boldsymbol{f}, X|Y) \propto p(Y|\boldsymbol{f}, X)p(\boldsymbol{f}|X, \boldsymbol{\theta})p(X)$ is intractable. This intractability is side-stepped by introducing a variational distribution over unknowns $(\boldsymbol{f}, X)$ augmented with $M$ inducing variables $\boldsymbol{u} \equiv \{\boldsymbol{u}_d\}_{d=1}^{D}$, each distributed with a GP prior $p(\boldsymbol{u}_d) \sim \mathcal{N}(\mathbf{0}, K_{mm})$. $K_{mm}$ takes as input inducing input locations $Z \in \mathbb{R}^{M \times Q}$ which live in latent space, are shared across dimensions and have dimensionality $Q$ (matching $\boldsymbol{x}_n$)(Titsias, 2009). The formulation

$$q(\boldsymbol{f}, X, \boldsymbol{u}) = \Big[\prod_{d=1}^{D} p(\boldsymbol{f}_d|\boldsymbol{u}_d, X)q(\boldsymbol{u}_d)\Big]q(X) \approx p(\boldsymbol{f}, X, \boldsymbol{u}|Y) \tag{2}$$

where $q(\boldsymbol{u}_d)$ is the variational distribution over the inducing variables and $q(X) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \mu_n, s_n \mathbb{I}_Q)$ admits a tractable lower bound to the marginal likelihood $p(Y|\boldsymbol{\theta})$ for specific choices of kernel functions. The variational formulation above gives rise to an evidence lower bound (ELBO) written in rudimentary form (full derivation enclosed in supplementary),

$$\mathcal{L}_{1:D} = \sum_{d=1}^{D} \mathcal{L}_d = \overbrace{\sum_{d=1}^{D}\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{f}, X, \boldsymbol{u})}[\log p(y_{n,d}|\boldsymbol{f}_d, \boldsymbol{x}_n)]}^{\mathcal{L}_1} - \text{KL}(q(X)||p(X)) - \text{KL}(\prod_{d=1}^{D} q(\boldsymbol{u}_d)|| \prod_{d=1}^{D} p(\boldsymbol{u}_d)), \tag{3}$$

where KL$(\cdot)$ denotes Kullback-Leibler divergence and can be computed in closed form if the variational distributions $q(X)$ and $q(\boldsymbol{u})$ are chosen to be Gaussian[2]. In fact, the $q(\boldsymbol{u}_d)$s are unrestricted and an analytic maximisation of the bound w.r.t $q(\boldsymbol{u}_d)$s turns out to yield Gaussian distributions as the optimal form. The final bound in Titsias and Lawrence (2010) is obtained by collapsing the bound by plugging in the optimal $q^*(\boldsymbol{u}_d) = \underset{q(\boldsymbol{u}_d)}{\arg\max}\, \mathcal{L}_d \; \forall d's$.

In this work we take a different approach, we use the uncollapsed version of the lower bound (with an explicit representation of $q(\boldsymbol{u}_d)$) enabling stochastic gradient methods and parallelised mini-batching of gradients for a truly scalable solution.

### 2.1. SVI for Bayesian GPLVM

Stochastic Variational Inference (SVI) (Hoffman et al., 2013) is used to scale variational inference to massive datasets by sub-sampling data to compute noisy gradients. The main prerequisite is that the ELBO should factorise across the data points. Another detail is the existence of *global* variables that make the observations conditionally independent given their local latent variable. The introduction of inducing variables $\boldsymbol{u}$ satisfies this criteria for variational

[1] $X$ appears non-linearly inside the covariance matrix $K_d$

[2] The priors $p(X)$ and $p(\boldsymbol{u})$ are chosen to be factorised Gaussian in the model set-up

sparse GP regression models, Hensman et al. (2013) derives the uncollapsed lower bound for the regression case with a 1d output, here we show the final form of the uncollapsed lower bound for the latent variable model setting with $D$-dimensional outputs $\boldsymbol{y}_d$ by expanding $\mathcal{L}_1$.

$$\mathcal{L}_{1:D} = \sum_{n,d} \log\mathcal{N}(y_{n,d}| \underbrace{\langle k_n^T \rangle_{q(X)}}_{\Psi_1^{(n,\cdot)}} K_{mm}^{-1}\boldsymbol{m}_d, \sigma_y^2) - \frac{1}{2\sigma_y^2}\text{Tr}(\underbrace{\langle K_{nn}\rangle_{q(X)}}_{\psi_0} - K_{mm}^{-1}\underbrace{\langle K_{mn}K_{nm}\rangle_{q(X)}}_{\Psi_2})$$
$$- \frac{1}{2\sigma_y^2}\text{Tr}(S_d K_{mm}^{-1}\underbrace{\langle K_{mn}K_{nm}\rangle_{q(X)}}_{\Psi_2} K_{mm}^{-1}) - \sum_n \text{KL}(q(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)) - \sum_d \text{KL}(q(\boldsymbol{u}_d)||p(\boldsymbol{u}_d))$$

This bound is optimised with respect to variational parameters of $q(\{\boldsymbol{u}\}_{d=1}^D)$ given by $(\{\boldsymbol{m}_d, S_d\}_{d=1}^D, Z)$, the variational parameters concerning $q(X)$ given by $(\{\boldsymbol{\mu}_n, s_n\}_{n=1}^N)$, the kernel hyperparameters $\boldsymbol{\theta}$ and the likelihood noise variance $\sigma_y^2$. Please refer to the appendix A.3 for a derivation of this bound and appendix A.4 for the $\Psi$ terms where we show how this formulation preserves factorisability.

## 3. Gaussian Process Latent Variable Flows

One of the advantages of the generalised SVI framework is that it only pre-requisites a factorisation of the approximating variational family and does not impose any restrictions over its form. As such, the choice of approximating variational family $q(X)$ can be flexible as long as we can sample from it and compute gradients w.r.t its parameters.In this class of models we make $q(X)$ generic by allowing the base Gaussian distribution around each latent point to be transformed by a sequence of normalising flows. The motivation for this is that the posterior distribution $p(X|Y)$ can be arbitrarily complex manifesting in multiple-modes and non-linear correlations in latent space. Further, the pathologies in the posterior distribution over the latent variables $X$ can be more pronounced in high-dimensional missing data settings.

Normalising flows (Rezende and Mohamed, 2015) leverage the fundamental rule for specifying probability densities of transformed random variables. When a random variable $z^{(0)} \sim p(z^{(0)})$ is transformed by a sequence of $k$ invertible and differentiable mappings composed together, the resulting random variable $g_k \circ g_{k-1} \circ \ldots g_1(z^{(0)}) = z^{(k)}$ has a density given by, $p(z^{(k)}) = p(z^{(0)})\left|\det \prod_{j=1}^k \dfrac{\partial g_j}{\partial z^{(j-1)}}\right|^{-1}$.

The variational distribution of the Bayesian GPLVM with a transformed Gaussian distribution is given by,

$$q(X) = \prod_{n=1}^N \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}_n, s_n\mathbb{I}_Q)\left|\det \prod_{j=1}^k \frac{\partial g_j}{\partial \boldsymbol{x}_n^{(j-1)}}\right|^{-1}, \tag{4}$$

where the parameters of the flow mappings $g_j$ are collected in $\lambda$, a set of additional variational parameters. We call this model *Gaussian Process latent Variable Flows*. It's main attributes are:

- The base Gaussian distribution is channeled through a finite sequence of normalising flows yielding an arbitrarily complex marginal distribution over the latent points $\boldsymbol{x}_n$
- It is the marginal densities $q(\boldsymbol{x}_n)$ that are warped by the flow transformations rather than the joint, yielding a joint density that still factorises across data points.
- This formulation preserves the factorisability of the bound (it can be written as a sum of $N$ terms) but with a richer non-Gaussian variational approximation. The parameters $\lambda$ of the $Q$ dimensional flow are shared between the data points enabling amortised inference.

We perform Monte Carlo expectations of the terms in the uncollapsed lower bound that involve $q(X)$ by sampling from the base Gaussian at each step and pushing them through the flow $g_k \circ g_{k-1} \circ \ldots g_1(\boldsymbol{x}^{(0)}) = \boldsymbol{x}^k$ to yield the final latent point. We do this for all $N$ across each dimension $Q$.

### 3.1. Missing Data Framework

In this section we summarise a framework for dealing with massively missing data at test time. The SVI objective factorises across data points and dimensions. The crux of the training procedure relies on the marginalisation principle of Gaussian distributions. More concretely, we can marginalise out the missing dimensions $\boldsymbol{y}_a$ as long as our data point $\boldsymbol{y}$ is modelled as a joint Gaussian. For a single data point split by its unobserved and observed dimensions,

$$\int \prod_{d\in a} \prod_{d\in o} p(\boldsymbol{y}_a, \boldsymbol{y}_o | \boldsymbol{u}_d, X) d\boldsymbol{y}_a = \prod_{d\in o} p(\boldsymbol{y}_o | \boldsymbol{u}_d, X), \tag{5}$$

where $a$ and $o$ denote the indices of missing and observed dimensions respectively with the full set of dimensions given as, $D = a \cup o$. $\boldsymbol{u}_d \in \mathbb{R}^M \forall d = 1, \dots, D$ denote the inducing variables which ensure conditional independence. This set-up reflects real-world data very closely which is often sparse with many missing and few overlapping dimensions.

## 4. Experiments

### 4.1. Missing MNIST

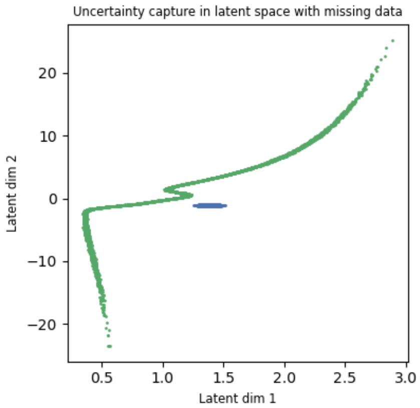

Figure 1: The latent flow based distribution (green) vs. the Gaussian distribution (blue) of the latent encoding of a single digit with missing pixels.

The focus of this experiment is to gauge how the models capture uncertainty when training with missing pixels. We use 5000 training samples from the MNIST digits dataset (LeCun et al., 2010) across classes [0-9] with ≈ 40% of the pixels missing in each digit. Fig. 3 summarises sample generation from the 2d latent distribution for a single training digit. The key feature of the flow based distribution is that it demonstrates the ability to capture non-linear correlations in the uncertainty structure of the latent posterior for the missing pixels digit. Notice that the samples generated from the flow based variational approximation are much more diverse than the ones from the latent Gaussian approximation; in some parts of the flow distribution they recover digits resembling the ground truth. Missing dimensions in the training data are highly likely to give rise to pathological posteriors with high uncertainty, and in these cases a flexible variational distribution provides a better approximation than a Gaussian. Fig. 2 shows samples generated from the flow based approximation, each point denotes a MAP estimate of the flow and we superimpose the reconstructed digits for some of the samples.

### 4.2. Movie Lens 100K

The movie lens 100K data has 1682 movies (columns/dimensions) across 943 users (rows/data points) where each user has rated an average of 20 movies (Harper and Konstan, 2015). This yields an extremely sparse data grid with 93.8% of the entries missing[3], truly embodying the massively missing data framework. We learn a 10$d$ latent distribution for the movie lens data and summarise the test RMSE and test log likelihoods for increasing flow lengths. We trained on 843 users and made predictions for 100 users.

| $q(X)$ | Test RMSE |
|---|---|
| $\prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \mu_n, s_n \mathbb{I}_Q)$ | 0.9648 |
| $Planar_{20}$ | 0.9281 |
| $Planar_{30}$ | 0.9105 |
| $Planar_{50}$ | 0.9039 |

Table 2: Movie Lens 100K RMSE summary with 10 latent dimensions. The planar flows of increasing lengths transform a base Gaussian distribution.

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

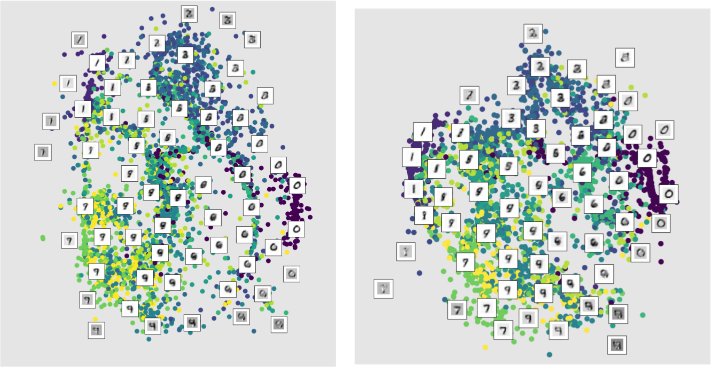

Figure 2: Sampled MNIST digits from the flow based (left) and Gaussian (right) 2d variational approximation trained on data with missing pixels. Both models have not seen a fully dense image of any digit.

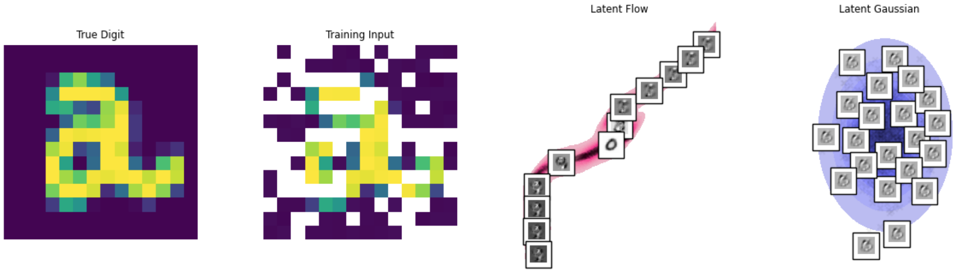

Figure 3: *Leftmost:* True down scaled MNIST digit (14×14). *Left:* Training input of the digit with 40% missing pixels. *Right:* Multiple samples from the flow based 2d latent distribution. *Rightmost:* Multiple samples from the Gaussian 2d latent distribution. The densities shown are for a single digit.

## 5. Further work & Extensions

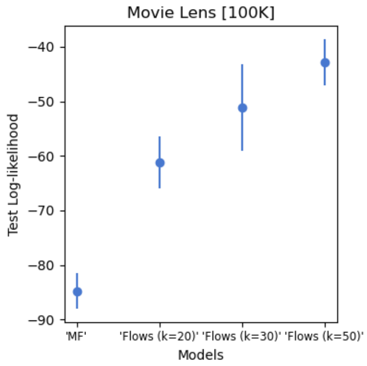

Figure 4: Test log-likelihood (higher is better) across the mean-field Gaussian (MF) model, and planar flows of different lengths. The training was done on a slice of the data (selecting movies with genre 'Sci_Fi' or 'Romance' )

The ability to model the distribution of a latent encoding flexibly could be invaluable in several settings. Flows enable capturing correlations in the latent space without paying the price of learning a full covariance matrix. In the missing training data framework, their benefit is even more pronounced as unsupervised training with massively missing dimensions induce arbitrarily complex posteriors in latent space. The implementation shown here preserves the factorisability of the lower bound enabling parallelisable inference achieving dual goals of scalability and flexibility. The highlight of the work is its applicability in massively missing data regimes which very few frameworks can handle. Further, there are two questions that surround the use of normalising flows, 1) the choice of flow and 2) the flow length. Different transformations yield flows with different inductive biases, this can be studied in more detail. Based on prior knowledge, we can introduce transformations which produce a desired effect in the latent space. Other avenues for theoretical work include carefully comparing the collapsed bound proposed in (Titsias and Lawrence, 2010) with the uncollapsed bound introduced in (Hensman et al., 2013) and derived here.

## References

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

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

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

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

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

Neil D. Lawrence and Joaquin 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.

Yann LeCun, Corinna Cortes, and Christopher J. Burges. Mnist handwritten digit database. 2010. *URL http://yann. lecun. com/exdb/mnist*, 7:23, 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes in machine learning*. Springer, 2006.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

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

## Appendix A. Derivations

### A.1. Motivation for inducing variables

As is standard, we wish to minimize the KL divergence between the variational approximation and the true posterior given by, $\mathrm{KL}(q(\{\boldsymbol{f}_d\}_{d=1}^{D}, X)||p(\{\boldsymbol{f}\}_{d=1}^{D}, X|Y))$. For an arbitrary $d$ we have,

$$KL(q(\boldsymbol{f}_d, X)||p(\boldsymbol{f}_d, X|Y)) = \int q(\boldsymbol{f}_d, X) \log \frac{q(\boldsymbol{f}_d, X)}{p(\boldsymbol{f}_d, X|Y)} d\boldsymbol{f}_d dX \tag{6}$$

$$= -\underbrace{\int q(\boldsymbol{f}_d, X) \log \frac{p(Y|\boldsymbol{f}_d, X)p(\boldsymbol{f}_d|X, \boldsymbol{\theta})p(X)}{q(\boldsymbol{f}_d, X)} d\boldsymbol{f}_d dX}_{ELBO} + \log p(Y|\boldsymbol{\theta}) \tag{7}$$

The evidence lower bound shown above is mathematically intractable due to the term $p(\boldsymbol{f}_d|X, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, K_{nn}^{(d)})$ involving the variables $X$ which appear non-linearly in the kernel matrix. It turns out that an augmented bound with inducing variables $\boldsymbol{u}_d$ for each dimension is tractable.

### A.2. Derivation of the ELBO eq. (3)

In this section we detail the derivation of the rudimentary ELBO as in eq. (3) of the paper,
We introduce auxiliary inducing variables, $\boldsymbol{u}_d \in \mathbb{R}^M$ for each of the latent functions $\boldsymbol{f}_d$, collecting all of the $\boldsymbol{u}_d$'s in $\boldsymbol{u}$ and $\boldsymbol{f}_d$'s in $\boldsymbol{f}$ for ease of notation, it turns out that variational inference in $(\boldsymbol{f}, \boldsymbol{u}, X)$ space is actually tractable. The marginal prior $p(\boldsymbol{u}|Z)$ is given as,

$$p(\boldsymbol{u}|Z) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{u}_d; \boldsymbol{0}, K_{mm}). \tag{8}$$

where $Z \in \mathbb{R}^{M \times Q}$ denote inducing input locations.
Writing down the augmented variational approximation as,

$$q(\boldsymbol{f}, \boldsymbol{u}, X) = \prod_{d=1}^{D} [p(\boldsymbol{f}_d|\boldsymbol{u}_d, X)q(\boldsymbol{u}_d)]q(X) \approx p(\boldsymbol{f}, \boldsymbol{u}, X|Y) \tag{9}$$

we derive the form of the ELBO in this augmented space by writing down the KL divergence explicitly,

$$\begin{aligned} KL(q(\boldsymbol{f}, \boldsymbol{u}, X)||p(\boldsymbol{f}, \boldsymbol{u}, X|Y)) &= \int p(\boldsymbol{f}|\boldsymbol{u}, X)q(\boldsymbol{u})q(X) \log \frac{p(\boldsymbol{f}|\boldsymbol{u}, X)q(\boldsymbol{u})q(X)}{p(\boldsymbol{f}, \boldsymbol{u}, X|Y)} d\boldsymbol{f} d\boldsymbol{u} dX \\ &= -\int p(\boldsymbol{f}|\boldsymbol{u}, X)q(\boldsymbol{u})q(X) \log \frac{p(Y|\boldsymbol{f}, X)p(\boldsymbol{f}|\boldsymbol{u}, X)p(\boldsymbol{u}|Z)p(X)}{p(\boldsymbol{f}|\boldsymbol{u}, X)q(\boldsymbol{u})q(X)} d\boldsymbol{f} d\boldsymbol{u} dX + \log p(Y) \end{aligned}$$

where the terms in red (the difficult terms) cancel out and we suppress the implicit conditioning over the hyperparameters $\boldsymbol{\theta}$ for brevity. The final ELBO is given by,

$$\mathcal{L}_{1:D} = p(\boldsymbol{f}|\boldsymbol{u}, X)q(\boldsymbol{u})q(X) \log \frac{p(Y|\boldsymbol{f}, X)p(\boldsymbol{u}|Z)p(X)}{q(\boldsymbol{u})q(X)} d\boldsymbol{f} d\boldsymbol{u} dX \tag{10}$$

where $p(\boldsymbol{f}|\boldsymbol{u}, X)$ is the conditional prior, $q(\boldsymbol{u})$ is a free-form variational distribution, $p(X) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{0}, \mathbb{I}_Q)$ is just a product of standard normals and $q(X)$ is chosen to be a product of multivariate Gaussians with diagonal covariances $s_n$,

$$q(X) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \mu_n, s_n \mathbb{I}_Q), \tag{11}$$

$p(\boldsymbol{u}|Z)$ is as defined in eq. 8.
Going back to (10), we first apply the integration w.r.t $\boldsymbol{f}$ and rewrite it more conveniently,

$$\begin{aligned} \mathcal{L}_{1:D} &= \int q(X) \left[ \int q(\boldsymbol{u}) \left[ \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u}, X)}[\log p(Y|\boldsymbol{f}, X)] + \log \frac{p(\boldsymbol{u})}{q(\boldsymbol{u})} + \log \frac{p(X)}{q(X)} \right] d\boldsymbol{u} \right] dX & (12) \\ &= \mathbb{E}_{q(\boldsymbol{f}, \boldsymbol{u}, X)}[\log p(Y|\boldsymbol{f}, X)] - KL(q(X)||p(X)) - KL(q(\boldsymbol{u})||p(\boldsymbol{u})) & (13) \\ &= \mathbb{E}_{q(\boldsymbol{f}, \boldsymbol{u}, X)}[\log \prod_{n=1}^{N} \prod_{d=1}^{D} \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2)] - KL(q(X)||p(X)) - KL(q(\boldsymbol{u})||p(\boldsymbol{u})) & (14) \\ &= \overbrace{\sum_{n,d} \mathbb{E}_{q(\boldsymbol{f}, \boldsymbol{u}, X)}[\log \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2)]}^{\mathcal{L}_1} - KL(q(X)||p(X)) - KL(q(\prod_{d=1}^{D} \boldsymbol{u}_d)|| \prod_{d=1}^{D} p(\boldsymbol{u}_d)) & (15) \end{aligned}$$

## A.3. Derivation of the uncollapsed SVI ELBO

Consider the first term of the ELBO in eq. 15,

$$
\begin{aligned}
\mathcal{L}_1 &= \sum_{n,d} \mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)q(\boldsymbol{u}_d)q(X)}[\log p(y_{n,d}|\boldsymbol{f}_d,\boldsymbol{x}_n)] \qquad (16)\\
&= \sum_{n,d} \int q(X) \int q(\boldsymbol{u}_d) \underbrace{\int p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)\log p(y_{n,d}|\boldsymbol{f}_d,\boldsymbol{x}_n)d\boldsymbol{f}_d}_{\mathcal{L}_f^{(n,d)}} d\boldsymbol{u}_d dX\\
&= \sum_{n,d} \int q(X) \underbrace{\int q(\boldsymbol{u}_d)\, \mathcal{L}_f^{(n,d)} d\boldsymbol{u}_d}_{\mathcal{L}_u^{(n,d)}} dX\\
&= \sum_{n,d} \underbrace{\int q(X)\, \mathcal{L}_u^{(n,d)} dX}_{\mathcal{L}_X^{(n,d)}}.
\end{aligned}
$$

First, performing the integration w.r.t $\boldsymbol{f}_d$,

$$
\begin{aligned}
\mathcal{L}_f^{(n,d)} &= \int p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)\log p(y_{n,d}|\boldsymbol{f}_d,\boldsymbol{x}_n)d\boldsymbol{f}_d \qquad (17)\\
&= \int p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)\Big[-\frac{1}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}(y_{n,d}^2 - 2y_{n,d}\boldsymbol{f}_d(\boldsymbol{x}_n) + (\boldsymbol{f}_d(\boldsymbol{x}_n))^2)\Big]d\boldsymbol{f}_d \qquad (18)\\
&= -\frac{1}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}(y_{n,d}^2) + \frac{y_{n,d}}{\sigma_y^2}\underbrace{\int \boldsymbol{f}_d(\boldsymbol{x}_n)p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)d\boldsymbol{f}_d}_{k_n^T K_{mm}^{-1}\boldsymbol{u}_d} - \frac{1}{2\sigma_y^2}\underbrace{\int(\boldsymbol{f}_d(\boldsymbol{x}_n))^2 p(\boldsymbol{f}_d|\boldsymbol{u}_d,X)d\boldsymbol{f}_d}_{q_{n,n}+(k_n^T K_{mm}^{-1}\boldsymbol{u}_d)^T(k_n^T K_{mm}^{-1}\boldsymbol{u}_d)} \qquad (19)\\
&= \log\mathcal{N}(y_{n,d}|k_n^T K_{mm}^{-1}\boldsymbol{u}_d,\sigma_y^2) - \frac{1}{2\sigma_y^2}q_{n,n}. \qquad (20)
\end{aligned}
$$

Note: $y_{n,d}$ is a scalar ($d^{th}$ dimension of point $y_n$), $k_n^T$ is a $1 \times M$ matrix - the $n^{th}$ row of $K_{nm}$, we know that $p(\boldsymbol{f}_d|\boldsymbol{u}_d,X) = \mathcal{N}(K_{nm}K_{mm}^{-1}\boldsymbol{u}_d, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn})$. Further, $\boldsymbol{f}_d(\boldsymbol{x}_n)$ is a scalar, denoting the value at index $\boldsymbol{x}_n$ of the vector $\boldsymbol{f}_d$. $q_{n,n}$ is the $n^{th}$ entry in the diagonal of matrix $Q_{nn} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$.

Then, performing the integration w.r.t $\boldsymbol{u}_d$ (we parameterise $q(\boldsymbol{u}_d) = \mathcal{N}(\boldsymbol{m}_d, S_d)$ as we know its optimal form is a Gaussian and using similar identities as above),

$$
\begin{aligned}
\mathcal{L}_u^{(n,d)} &= \int q(\boldsymbol{u}_d)\Big[\log\mathcal{N}(y_{n,d}|k_n^T K_{mm}^{-1}\boldsymbol{u}_d,\sigma_y^2) - \frac{1}{2\sigma_y^2}q_{n,n}\Big]d\boldsymbol{u}_d\\
&= \log\mathcal{N}(y_{n,d}|k_n^T K_{mm}^{-1}\boldsymbol{m}_d,\sigma_y^2) - \frac{1}{2\sigma_y^2}q_{n,n} - \frac{1}{2\sigma_y^2}\Lambda_{n,n}.
\end{aligned}
\qquad (21)
$$

where $\Lambda_{n,n}$ is the $n^{th}$ entry of the diagonal of matrix $\Lambda = S_d K_{mm}^{-1} K_{mn} K_{nm} K_{mm}^{-1}$. Now, what remains is to perform the integration w.r.t $q(X)$.

$$\begin{aligned}\mathcal{L}_1 = \sum_{n,d} \mathcal{L}_X^{(n,d)} = \sum_{n,d} \log \mathcal{N}(y_{n,d} | \underbrace{\langle k_n^T \rangle_{q(X)}}_{\Psi_1^{(n,\cdot)}} K_{mm}^{-1} \boldsymbol{m}_d, \sigma_y^2) - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\underbrace{\langle K_{nn} \rangle_{q(X)}}_{\psi_0} - K_{mm}^{-1} \underbrace{\langle K_{mn} K_{nm} \rangle_{q(X)}}_{\Psi_2}) \\ - \frac{1}{2\sigma_y^2} \mathrm{Tr}(S_d K_{mm}^{-1} \underbrace{\langle K_{mn} K_{nm} \rangle_{q(X)}}_{\Psi_2} K_{mm}^{-1})\end{aligned}$$

We note that the only terms in $\mathcal{L}_u^{(n,d)}$ involving the latent $X$ are $K_{nm}$, $K_{nn}$ and $K_{nm}K_{mn}$;we show in the next section that they admit a factorisation across data points. The final bound is given by,

$$\begin{aligned}\mathcal{L}_{1:D} &= \mathcal{L}_1 - \sum_n \mathrm{KL}(q(\boldsymbol{x}_n) || p(\boldsymbol{x}_n)) - \sum_d \mathrm{KL}(q(\boldsymbol{u}_d) || p(\boldsymbol{u}_d)) \qquad (22) \\ &= \sum_{n,d} \log \mathcal{N}(y_{n,d} | \underbrace{\langle k_n^T \rangle_{q(X)}}_{\Psi_1^{(n,\cdot)}} K_{mm}^{-1} \boldsymbol{m}_d, \sigma_y^2) - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\underbrace{\langle K_{nn} \rangle_{q(X)}}_{\psi_0} - K_{mm}^{-1} \underbrace{\langle K_{mn} K_{nm} \rangle_{q(X)}}_{\Psi_2}) \\ &- \frac{1}{2\sigma_y^2} \mathrm{Tr}(S_d K_{mm}^{-1} \underbrace{\langle K_{mn} K_{nm} \rangle_{q(X)}}_{\Psi_2} K_{mm}^{-1}) - \sum_n \mathrm{KL}(q(\boldsymbol{x}_n) || p(\boldsymbol{x}_n)) - \sum_d \mathrm{KL}(q(\boldsymbol{u}_d) || p(\boldsymbol{u}_d)) \\ & \qquad (23)\end{aligned}$$

### A.4. $\Psi$ statistics

In this section we detail how $\psi_0$, $\Psi_1$ and $\Psi_2$ are factorisable across data points.

$$\begin{aligned}\psi_0 &= \mathrm{Tr}(\langle K_{nn} \rangle_{q(X)}) && (24) \\ &= \left\langle \sum_{i=1}^{N} K_{nn}^{(ii)} \right\rangle_{q(X)} \quad \text{where} K_{nn}^{(ii)} \text{ are the diagonal entries of matrix } K_{nn} && (25) \\ &= \sum_{i=1}^{N} \langle K_{nn}^{(ii)} \rangle_{q(\boldsymbol{x}_i)} \quad \text{where } \{\boldsymbol{x}_i\}_{i=1}^{N} \equiv X \text{ and } q(X) = \prod_{i=1}^{N} q(\boldsymbol{x}_i) && (26)\end{aligned}$$

Next, we look at $\Psi_1$,

$$\Psi_1 = \langle K_{nm} \rangle_{q(X)} \tag{27}$$

$$K_{nm} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{z}_M) \\ k(\boldsymbol{x}_2, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_2, \boldsymbol{z}_M) \\ \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{z}_M) \end{bmatrix} = \begin{bmatrix} \text{——} & K_{nm}^{(1,\cdot)} & \text{——} \\ \text{——} & K_{nm}^{(2,\cdot)} & \text{——} \\ \vdots & \vdots & \vdots \\ \text{——} & K_{nm}^{(N,\cdot)} & \text{——} \end{bmatrix} \tag{28}$$

$$\Psi_1 = \begin{bmatrix} \text{——} & \Psi_1^{(1,\cdot)} & \text{——} \\ \text{——} & \Psi_1^{(2,\cdot)} & \text{——} \\ \vdots & \vdots & \vdots \\ \text{——} & \Psi_1^{(N,\cdot)} & \text{——} \end{bmatrix} = \begin{bmatrix} \text{——} & \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \text{——} \\ \text{——} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \text{——} \\ \vdots & \vdots & \vdots \\ \text{——} & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} & \text{——} \end{bmatrix}, \tag{29}$$

where we notice that $\Psi_1$ is a $N \times M$ matrix where each row just depends on a data point $\boldsymbol{x}_i$.

$$
\begin{align}
\Psi_2 &= \langle K_{mn} K_{nm} \rangle_{q(X)} \tag{30} \\
&= \begin{bmatrix} \Big| & \Big| & \vdots & \Big| \\ \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \dots & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} \\ \Big| & \Big| & \vdots & \Big| \end{bmatrix} \begin{bmatrix} \text{——} & \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \text{——} \\ \text{——} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \text{——} \\ \vdots & \vdots & \vdots \\ \text{——} & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} & \text{——} \end{bmatrix} \tag{31} \\
&= \sum_{i=1}^{N} \langle K_{nm}^{(i,\cdot)^T} K_{nm}^{(i,\cdot)} \rangle_{q(\boldsymbol{x}_i)} \tag{32}
\end{align}
$$

which is an $M \times M$ matrix decomposable as a sum of $N$ $M \times M$ matrices where each component matrix is only dependent on a data point $\boldsymbol{x}_i$.

### A.5. KL divergence between factorised Gaussians

In eq. (9) we re-write the KL term involving $q(X)$ as a factorisation across $n$, we show the proof below:

$$
\begin{aligned}
\mathrm{KL}(q(X)||p(X)) &= \mathrm{KL}\Big( \prod_{n=1}^{N} q(\boldsymbol{x}_n) || \prod_{n=1}^{N} p(\boldsymbol{x}_n) \Big) \\
&= \int \prod_{n=1}^{N} q(\boldsymbol{x}_n) \log \frac{\prod_{n=1}^{N} q(\boldsymbol{x}_n)}{\prod_{n=1}^{N} p(\boldsymbol{x}_n)} d\boldsymbol{x}_1 \dots d\boldsymbol{x}_N \\
&= \int \prod_{n=1}^{N} q(\boldsymbol{x}_n) \sum_{n=1}^{N} \log \frac{q(\boldsymbol{x}_n)}{p(\boldsymbol{x}_n)} d\boldsymbol{x}_1 \dots d\boldsymbol{x}_N \\
&= \int \prod_{n=1}^{N-1} q(\boldsymbol{x}_n) q(\boldsymbol{x}_N) \Big( \log \frac{q(\boldsymbol{x}_N)}{p(\boldsymbol{x}_N)} + \sum_{n=1}^{N-1} \log \frac{q(\boldsymbol{x}_n)}{p(\boldsymbol{x}_n)} \Big) d\boldsymbol{x}_1 \dots d\boldsymbol{x}_N \\
&= \mathrm{KL}(q(\boldsymbol{x}_N)||p(\boldsymbol{x}_N)) \underbrace{\int \prod_{n=1}^{N-1} q(\boldsymbol{x}_n) d\boldsymbol{x}_1 \dots d\boldsymbol{x}_{N-1}}_{1} + \mathrm{KL}\Big( \prod_{n=1}^{N-1} q(\boldsymbol{x}_n) || \prod_{n=1}^{N-1} p(\boldsymbol{x}_n) \Big) \\
&= \sum_{n=1}^{N} \mathrm{KL}(q(\boldsymbol{x}_n)||p(\boldsymbol{x}_n))
\end{aligned}
$$

### A.6. Test log-likelihood

1. The test log-likelihood is given by $\log p(\boldsymbol{y}^{*}|Y)$, we report this metric averaged across the number of test points $n_{test}$ and describe how its computed below.

This metric is sometimes called the log predictive density, they both refer to the same quantity. The test log-likelihood is given by $\log p(\boldsymbol{y^*}|Y)$,

$$\begin{aligned} p(\boldsymbol{y^*}|Y) &= \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{u})p(\boldsymbol{u}|Y)q(\boldsymbol{x}^*)d\boldsymbol{u}d\boldsymbol{x}^* \\ &\approx \frac{1}{J}\sum_{j=1}^{J}\int p(\boldsymbol{y}^*|\boldsymbol{x}^*_{(j)}, \boldsymbol{u})q^*(\boldsymbol{u})d\boldsymbol{u} \quad \boldsymbol{x}^*_{(j)} \sim q(\boldsymbol{x}^*) \\ &\approx \frac{1}{J}\sum_{j=1}^{J}\mathcal{N}(K_{*m}K_{mm}^{-1}\boldsymbol{m}, K_{**} - Q_{**} + K_{*m}K_{mm}^{-1}\Sigma K_{mm}^{-1}K_{m*}) \end{aligned} \tag{33}$$

where, $Q_{**} = K_{*m}K_{mm}^{-1}K_{m*}$,

$$\begin{aligned} q^*(\boldsymbol{u}) &= \mathcal{N}(\boldsymbol{m}, \Sigma) \\ \boldsymbol{m} &= \sigma_n^{-2}K_{mm}(K_{mm} + \sigma_n^{-2}K_{mn}K_{nm}) \\ \Sigma &= K_{mm}(K_{mm} + \sigma_n^{-2}K_{mn}K_{nm})^{-1}K_{mm} \end{aligned} \tag{34}$$

The integral over $\boldsymbol{x}^*$ is resolved numerically and the integral over $\boldsymbol{u}$ is analytical as per Titsias (2009).