Generalised Gaussian Process Latent Variable Models (GPLVM) with Stochastic Variational Inference

Vidhi Lalchand
University of Cambridge

Aditya Ravuri
University of Cambridge

Neil D. Lawrence
University of Cambridge

Abstract

Gaussian process latent variable models (GPLVM) are a flexible and non-linear approach to dimensionality reduction, extending classical Gaussian processes to an unsupervised learning context. The Bayesian incarnation of the GPLVM [Titsias and Lawrence, 2010] uses a variational framework, where the posterior over latent variables is approximated by a well-behaved variational family, a factorised Gaussian yielding a tractable lower bound. However, the non-factorisability of the lower bound prevents truly scalable inference. In this work, we study the doubly stochastic formulation of the Bayesian GPLVM model amenable with minibatch training. We show how this framework is compatible with different latent variable formulations and perform experiments to compare a suite of models. Further, we demonstrate how we can train in the presence of massively missing data and obtain high-fidelity reconstructions. 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* and 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. In the GPLVM the direction of the mapping is reversed.

The standard GPLVM is a multi-output regression model where 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) in a way that factorizes across the data enabling minibatching 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* or 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. This 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 latent space to be automatically determined by using the standard *automatic relevance determination* squared exponential (SE-ARD) kernel whose lengthscales are determined by maximisation of the ELBO. Extraneous dimensions acquire longer lengthscales and are automatically pruned. However, this closed form framework does not factorise across data points [Titsias and Lawrence, 2010] preventing the application of Bayesian GPLVM to larger datasets.

In this paper we extend the big data regression setting

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

proposed in Hensman et al. [2013] to the unsupervised latent variable model setting. We re-formulate Bayesian GPLVM for scalable inference using SVI by using a structured doubly stochastic lower bound [Titsias and Lázaro-Gredilla, 2014]. 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. Such a model was considered in Bui and Turner [2015] and this is the fourth model we include in our compendium which we call *Autoencoded Bayesian SVI* or AEB-SVI. In summary, our main contributions are:

- Present a generalised framework for GPLVM models which differ in the form of the latent variable set-up and 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), Bayesian SVI (B-SVI) and AEB-SVI models enabling efficient and scalable inference.
- Extend this framework to dimensionality reduction for non-conjugate likelihoods across all the latent variable incarnations.
- Demonstrate how training in these models is compatible with partially and massively missing data settings[1] frequently embodied in real-world datasets.

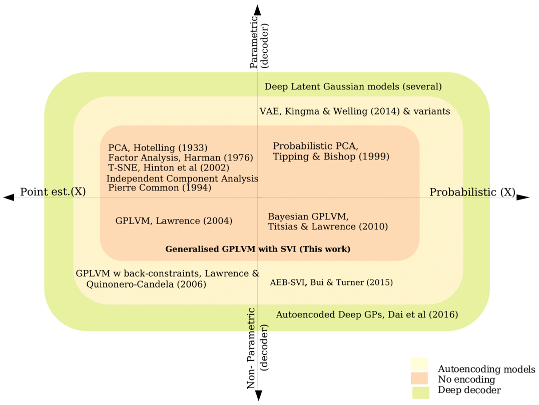

Parametric (decoder)
Non- Parametric (decoder)
Point est.(X)
Probabilistic (X)
Deep Latent Gaussian models (several)
VAE, Kingma & Welling (2014) & variants
PCA, Hotelling (1933)
Factor Analysis, Harman (1976)
T-SNE, Hinton et al (2002)
Independent Component Analysis
Pierre Common (1994)
Probabilistic PCA,
Tipping & Bishop (1999)
GPLVM, Lawrence (2004)
Bayesian GPLVM,
Titsias & Lawrence (2010)
Generalised GPLVM with SVI (This work)
GPLVM w back-constraints, Lawrence &
Quinonero-Candela (2006)
AEB-SVI, Bui & Turner (2015)
Autoencoded Deep GPs, Dai et al (2016)
Autoencoding models
No encoding
Deep decoder

Figure 1: A taxonomy of latent variable models for unsupervised dimensionality reduction along three axis of variation. 1) the form of the latent variable, 2) the nature of the decoder and 3) whether or not the models are autoencoding. The framework in this work is amenable with point estimation and Bayesian learning as well as amortisation.

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:

$$
\begin{aligned}
p(\mathbf{X}) &= \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \mathbf{0}, \mathbb{I}_Q), \\
p(\mathbf{F}|\mathbf{U}, \mathbf{X}, \boldsymbol{\theta}) &= \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{f}_d; K_{nm}K_{mm}^{-1}\boldsymbol{u}_d, Q_{nn}), \qquad (1) \\
p(\mathbf{Y}|\mathbf{F}, X) &= \prod_{n=1}^{N} \prod_{d=1}^{D} \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2),
\end{aligned}
$$

where $Q_{nn} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$, $\mathbf{F} \equiv \{\boldsymbol{f}_d\}_{d=1}^{D}$, $\mathbf{U} \equiv \{\boldsymbol{u}_d\}_{d=1}^{D}$ 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.

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,

$$
\begin{aligned}
p(\mathbf{F}, \mathbf{X}, \mathbf{U}|\mathbf{Y}) &= \Big[\prod_{d=1}^{D} p(\boldsymbol{f}_d|\boldsymbol{u}_d, X)q(\boldsymbol{u}_d)\Big]q(\mathbf{X}) \qquad (2) \\
&\approx q(\mathbf{F}, \mathbf{X}, \mathbf{U})
\end{aligned}
$$

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

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

Table 1: Existing approaches for Inference in GPLVMs. Our work studies the scalable alternative with SVI across all these models. The decoder ($X \longrightarrow Y$) is a GP across all methods.

Reference	Data Likelihood	Latent Variable $q(X)$	Encoder ($Y \rightarrow X$)	Training Method
Lawrence [2004]	Gaussian	point est.	✗	Gradient descent
Lawrence and Quiñonero Candela [2006]	Gaussian	point est.	✓	Gradient descent
Titsias and Lawrence [2010]	Gaussian	Gaussian	✗	Collapsed VI
Bui and Turner [2015]	Gaussian	Gaussian	✓	SVI
Ramchandran et al. [2021]	Any	Gaussian	✓	SVI
This work	Any	point / Gaussian	✗/✓	SVI

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. Beyond speeding up inference, the uncollapsed bound has properties which open up several possibilities, for instance, training with high-dimensional data with a non-Gaussian likelihood and structure discovery in the presence of sparse, high-dimensional data.

3 Generalised GPLVM with SVI

The key insight from Hensman et al. [2013] is to keep the representation of $\mathbf{U}$ uncollapsed and learn $q(\boldsymbol{u}_d) \sim \mathcal{N}(\boldsymbol{m}_d, S_d)$ numerically using stochastic gradient methods. In the next sections, we extend this insight to variationally learning $q(\mathbf{X})$.

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 for clarity, hence drop the dimension index d.

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

$$\triangleq \ln \tilde{p}(\boldsymbol{y}|\boldsymbol{u}) = \prod_{n=1}^{N} \mathcal{N}(y_n | k_n^T K_{mm}^{-1} \boldsymbol{u}, \sigma_y^2) \times$$

$$\exp\left\{ -\frac{1}{2\sigma_y^2} (k_{nn} - k_n^T K_{mm}^{-1} k_n) \right\} \tag{3}$$

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

3.2 Doubly Stochastic Evidence Lower bound (DS-ELBO)

The term *doubly stochastic inference* 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 goal is dimensionality reduction.

Keeping with the formulation in section 2.1 we write down the rudimentary ELBO, with the familiar decomposition involving the expected log-likelihood term and KL terms,

$$\mathcal{L} = \int p(\mathbf{F}|\mathbf{U}, \mathbf{X}) q(\mathbf{U}) q(\mathbf{X}) \log \frac{p(\mathbf{Y}|\mathbf{F}, \mathbf{X}) p(\mathbf{U}|Z) p(\mathbf{X})}{q(\mathbf{U}) q(\mathbf{X})} d\mathbf{F} d\mathbf{U} d\mathbf{X} \tag{4}$$

$$= \mathbb{E}_{q(.)}[\log p(\mathbf{Y}|\mathbf{F}, \mathbf{X})] - \text{KL}(q(\mathbf{X})||p(\mathbf{X})) - \text{KL}(q(\mathbf{U})||p(\mathbf{U}))$$

where $q(.)$ is as in eq. (2).

3.2.1 Analytical derivation of the factorised form: Gaussian and Non-Gaussian likelihoods

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 $\phi = \{\mu_n, s_n \mathbb{I}_Q\}$ and the variational distribution over the inducing variables as $q_\lambda(\boldsymbol{u}_d)$ where $\lambda = \{\boldsymbol{m}_d, S_d\}$.

$$\mathcal{L}(\mathcal{D}) = \mathbb{E}_{q(.)} \left[\sum_{n,d} \log \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2) \right] \tag{5}$$

$$\underbrace{- \sum_{n} \text{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n)) - \sum_{d} \text{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))}_{\text{KL terms}}$$

$$= \sum_{n,d} \mathbb{E}_{q_\phi(\boldsymbol{x}_n)} [\underbrace{\mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d, \boldsymbol{x}_n) q_\lambda(\boldsymbol{u}_d)} [\log \mathcal{N}(y_{n,d}; \boldsymbol{f}_d(\boldsymbol{x}_n), \sigma_y^2)]]}_{\mathcal{L}_{n,d}(\boldsymbol{x}_n, y_{n,d}) = \mathcal{L}_{n,d}}$$

$$- \text{KL terms}$$

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,

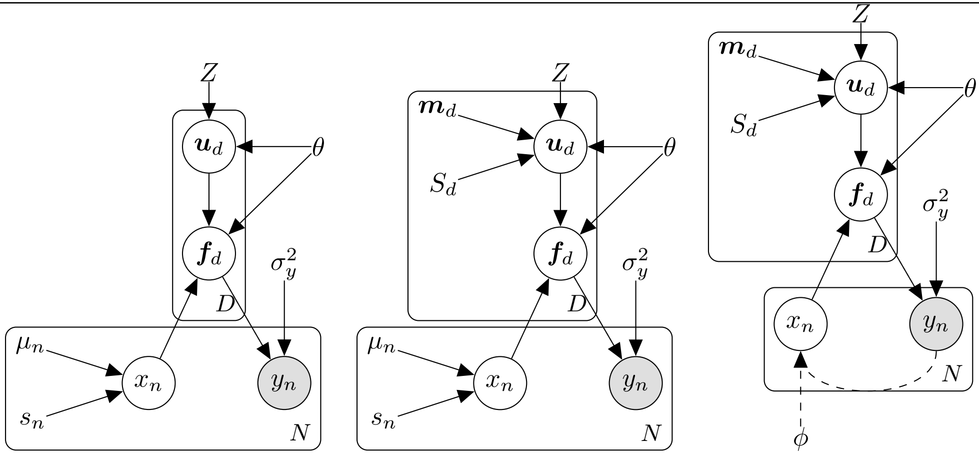


Figure 2: *Left:* Graphical model of the collapsed bound formulation of the Bayesian GPLVM. *Middle:* B-SVI where we learn individual parameters for each latent point. *Right:* AEB-SVI where the parameters for each latent point are deterministically derived by encoding the data point with the amortising neural network.

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_n)}[\mathcal{L}_{n,d}]$$
$$=\int q_\phi(\boldsymbol{x}_n)\int q_\lambda(\boldsymbol{u}_d)\int p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)\log\mathcal{N}(y_{n,d};\boldsymbol{f}_d(\boldsymbol{x}_n),\sigma_y^2)\, d\boldsymbol{f}_d(\boldsymbol{x}_n)d\boldsymbol{u}_d d\boldsymbol{x}_n$$
$$=\log\mathcal{N}(y_{n,d}|\underbrace{\langle K_{nm}\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_1^{(n,\cdot)}}K_{mm}^{-1}\boldsymbol{m}_d,\sigma_y^2)$$
$$-\frac{1}{2\sigma_y^2}\mathrm{Tr}(\underbrace{\langle K_{nn}\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\psi_0^n})+\frac{1}{2\sigma_y^2}\mathrm{Tr}(K_{mm}^{-1}\underbrace{\langle K_{mn}K_{nm}\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_2^n}) \qquad (6)$$
$$-\frac{1}{2\sigma_y^2}\mathrm{Tr}(S_dK_{mm}^{-1}\underbrace{\langle K_{mn}K_{nm}\rangle_{q_\phi(\boldsymbol{x}_n)}}_{\Psi_2^n}K_{mm}^{-1})$$

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

$$\Psi^{(n,\cdot)}\approx\frac{1}{J}\sum_{j=1}^{J}k(\boldsymbol{x}_n^{(j)},Z),\Psi_2^n\approx\frac{1}{J}\sum_{j=1}^{J}k(Z,\boldsymbol{x}_n^{(j)})k(\boldsymbol{x}_n^{(j)},Z),$$
$$\psi_0^n\approx\frac{1}{J}\sum_{j=1}^{J}k(\boldsymbol{x}_n^{(j)},\boldsymbol{x}_n^{(j)}) \qquad (7)$$

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 $\boldsymbol{x}_n^{(j)}=\mu_n+s_n\odot\epsilon^{(j)}$.

$$\mathbb{E}_{q_\phi(\boldsymbol{x}_n)}[\mathcal{L}_{n,d}]\simeq\frac{1}{J}\sum_{j=1}^{J}\mathcal{L}_{n,d}(\boldsymbol{x}_n^{(j)},y_{n,d})$$
$$\simeq\frac{1}{J}\sum_{j=1}^{J}\mathcal{L}_{n,d}(\mu_n+s_n\odot\epsilon^{(j)},y_{n,d}) \qquad (8)$$
$$=\frac{1}{J}\sum_{j=1}^{J}\mathcal{L}_{n,d}(g_\phi(\epsilon^{(j)}),y_{n,d})$$

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

$$\hat{\mathcal{L}}=\sum_n\sum_d\overbrace{\frac{1}{J}\sum_{j=1}^{J}\mathcal{L}_{n,d}(g_\phi(\epsilon^{(j)}),y_{n,d})}^{\hat{\mathcal{L}}_{n,d}} \qquad (9)$$
$$-\sum_d\mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))-\sum_n\mathrm{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n))$$

For a non-Gaussian likelihood (following on from eq. (6)), the expectations around the log-likehood term are intractable, instead one simplifies down to the marginals $q(\boldsymbol{f}_d|\boldsymbol{x}_n)$ analytically computable with standard Gaussian identities,

$$\int p(\boldsymbol{f}_d|\boldsymbol{u}_d,\boldsymbol{x}_n)q_\lambda(\boldsymbol{u}_d)d\boldsymbol{u}_d=q(\boldsymbol{f}_d|\boldsymbol{x}_n) \qquad (10)$$
$$=\mathcal{N}(k_n^TK_{mm}^{-1}\boldsymbol{m}_d,k_{nn}+k_n^TK_{mm}^{-1}(S_d-K_{mm})K_{mm}^{-1}k_n)$$

where k_n^T is the n^{th} row of K_{nm} only dependent on input $\boldsymbol{x}_n$ and $k_{nn}=k(\boldsymbol{x}_n,\boldsymbol{x}_n)$. Further, $q(\boldsymbol{f}_d|\boldsymbol{x}_n)$ denotes the marginal latent GP $\boldsymbol{f}_d$ conditioned at input $\boldsymbol{x}_n$. This gives the simplified lower bound,

$$\hat{\mathcal{L}}=\sum_{n,d}\mathbb{E}_{q(\boldsymbol{f}_d|\boldsymbol{x}_n)q_\phi(\boldsymbol{x}_n)}[\log p(y_{n,d}|\boldsymbol{f}_d(\boldsymbol{x}_n))] \qquad (11)$$
$$-\sum_d\mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))-\sum_n\mathrm{KL}(q_\phi(\boldsymbol{x}_n)||p(\boldsymbol{x}_n))$$

The POINT model in experiments comprises of just the first two terms in eq. (11), 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 (see Algorithm 1) 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 numerical expectations by sampling from the variational distributions $q(\boldsymbol{x}_n)$ and due to mini-batching for gradient updates.

3.3 Amortised Inference with Encoders

The GPLVM model provides a probabilistic non-linear mapping from latent space $\mathbf{X}$ to data space $\mathbf{Y}$, hence, 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]).

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 (parameterised through a factorization) per data-point thereby capturing correlations across dimensions (per latent point) in latent space.

$$q(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; G_{\phi_1}(\boldsymbol{y}_n), H_{\phi_2}(\boldsymbol{y}_n) H_{\phi_2}(\boldsymbol{y}_n)^T) \tag{12}$$

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. Bui and Turner [2015] present this model for a Gaussian likelihood setting.

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. 12). In the B-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 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 Computational Complexity

The training cost of the canonical non-SVI Bayesian GPLVM is dominated by $\mathcal{O}(NM^2D)$ where $M << N$ is the number of inducing variables and D is the data-dimensionality (we have D GP mappings $\mathbf{f}_d$ per output dimension), with the SVI framework this is reduced to $\mathcal{O}(M^3D)$ (free of N). The practical algorithm is made further scalable with the use of mini-batched learning admissable under the uncollapsed lower bound (this work). However the number of global variational parameters to be updated in each step (parameters of $q(\mathbf{U})$) is now increased. We summarise the number of global and local variational parameters across all the latent variable formulations in the table 2. The 'Canonical' model refers to Titsias and Lawrence [2010] and depends on the optimisation of MQ global parameters pertaining to the Q-dimensional inducing inputs Z. B-SVI on the other hand depends on $MQ + MD + M^2D$ parameters (inducing inputs, mean and dense covariance of inducing variables per latent dimension). The AEB-SVI model only has global parameters. The number of local variational parameters (parameters of $q(\mathbf{X})$) are the same between the canonical and B-SVI model at $2NQ$. At prediction time we need to learn the $2N^*Q$ local variational parameters from the augmented ELBO (for both the canonical and B-SVI model), this is further sped up in our framework with the AEB-SVI model which provides constant time $\mathcal{O}(1)$ test predictions.

3.6 Training with Many 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

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

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 $\sigma_y^2$),
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:
          $\hat{\mathcal{L}}(\mathcal{D}_B) = \frac{N}{B}\Big(\sum_b \sum_d \hat{\mathcal{L}}_{b,d} - \sum_b \mathrm{KL}(q_\phi(\boldsymbol{x}_b)||p(\boldsymbol{x}_b)\Big) - \sum_d \mathrm{KL}(q_\lambda(\boldsymbol{u}_d)||p(\boldsymbol{u}_d|Z))$
    • Gradient step: $Z, \boldsymbol{\theta}, \sigma_y^2, \{\mu_b, s_b\}_{b=1}^{B}\{m_d, S_d\}_{d=1}^{D} \longleftarrow$ optim($\hat{\mathcal{L}}(\mathcal{D}_B)$)
end
return $Z, \boldsymbol{\theta}, \phi, \lambda$
```

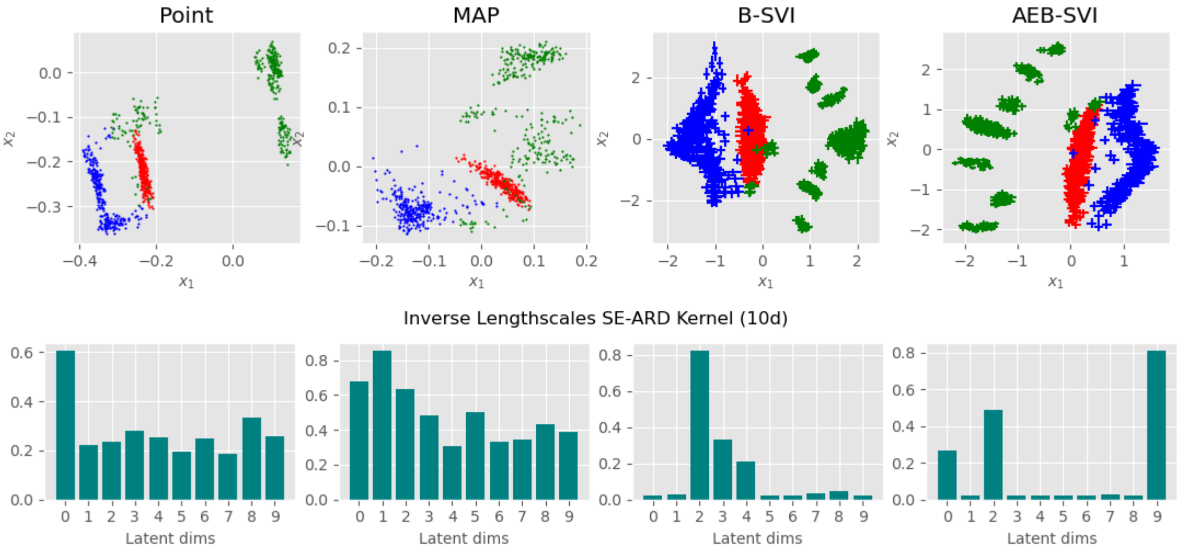


Figure 3: Top: The $2d$ latent space corresponding to the dominant dimensions learnt by each model. Bottom: The inverse lengthscales learnt by each model specification. We include a similar report for qPCR in the supplementary.

Table 2: Summary of compute across different models.

Model	Canonical	Point	MAP	B-SVI	AEB-SVI
Global (λ)	MQ	$MQ + MD + M^2D$			$MQ + \|\phi_1 + \phi_2\|$
Local	$2NQ$	NQ		$2NQ$	–

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,

$$\int \prod_{d \in a} \prod_{d \in o} p(\boldsymbol{y}_a, \boldsymbol{y}_o | \boldsymbol{u}_d, \mathbf{X}) d\boldsymbol{y}_a = \prod_{d \in o} p(\boldsymbol{y}_o | \boldsymbol{u}_d, \mathbf{X}) \quad (13)$$

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. 11 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 $q(\boldsymbol{x}) = \mathcal{N}(\mu^* s^* \mathbb{I}_Q)$.

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 B-SVI when faced with missing dimensions at training time. The missing data framework is not immediately compatible with auto-encoding models (AEB-SVI) as every latent point $\boldsymbol{x}_n$ is expressed as a function of the data point $\boldsymbol{y}_n$. However, set encoders [Qi et al., 2017; Vedantam et al., 2017; Ma et al., 2018] can be integrated as the auto-encoding component instead of a standard neural network. We defer this to future work.

4 Experiments

4.1 Ablation Study: Quantitative Results

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. We assess each model on their ability to reconstruct unseen high-dimensional points, automatic regularisation and detecting class structure in latent space. Further results and 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 also use a count dataset constructed from the NYC taxi cab records nyc.gov [2020] where we use vehicle counts of yellow/green/for-hire cabs aggregated by hour over the month of Jan 2020. We use a 80/20 split for training/testing and report test performance with ± 2 standard errors over three optimization runs. Since the training is unsupervised, the inherent ground-truth labels were not a part of training.

The 2d projections of the latent space (for oilflow data) 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 exhibiting automatic regularisation along with better test reconstruction errors.

The training/test error comparison (fig. 4) provides further evidence of overfitting in the point methods for high-dimensional datasets. The quality of the 2d latent projection of training data using the fully trained model might hide the overfitting effects as it is equally effective at disentangling the class structure.

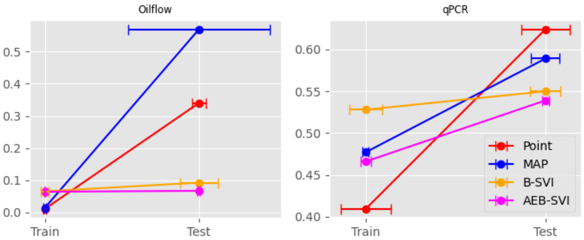


Figure 4: Left: The train and test RMSE per model showing evidence on overfitting for the non-Bayesian incarnations of Point and MAP.

For the taxi-cab data we use the likelihood, $p(\boldsymbol{y}_n|\boldsymbol{f}_n) = \prod_{d=1}^{D} \text{Poisson}(\exp(\boldsymbol{f}_d(\boldsymbol{x}_n)))$ and 10 samples from $q(\boldsymbol{x}_n)$ to approximate the expectation. All methods give very good test reconstructions (see supplementary for plots), however, it might seem like B-SVI and AEB-SVI underperform due to the higher RMSEs but the magnitude of the count values lead to larger variations in the test scores reported. An important factor is the dimensionality of the data, the benefit of the Bayesian techniques are subdued when acting on low-dimensional data as well as the importance of capturing correlations in latent space is more pronounced when the data has several dimensions. We show additional analysis and reconstructions 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.

4.2 Missing data: Reconstructing Structured Images & Human Motion

The focus of this experiment is to qualitatively assess how the models capture uncertainty when training with missing data in structured inputs. We use 15K training samples from the MNIST digits dataset [LeCun et al., 2010] with ≈ 60% of the pixels missing at random in each digit. Each image has 768 pixels yielding a 768d data space. The image data set [Roweis and Saul, 2000] contains ≈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. 5 summarises sample

Table 3: Test RMSE for datasets with ± 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	Likelihood	Z	Q	Point	MAP	B-SVI	AEB-SVI
Oilflow	1000 / 12	Gaussian	25	10	0.341 (0.008)	0.569 (0.092)	0.0925 (0.025)	**0.067 (0.0016)**
qPCR	450 / 48	Gaussian	40	11	0.624 (0.027)	0.589 (0.016)	**0.554 (0.017)**	**0.539 (0.004)**
Taxi-cab	744 / 3	Poisson	36	2	118 (21)	134 (11)	249 (81)	232 (22)

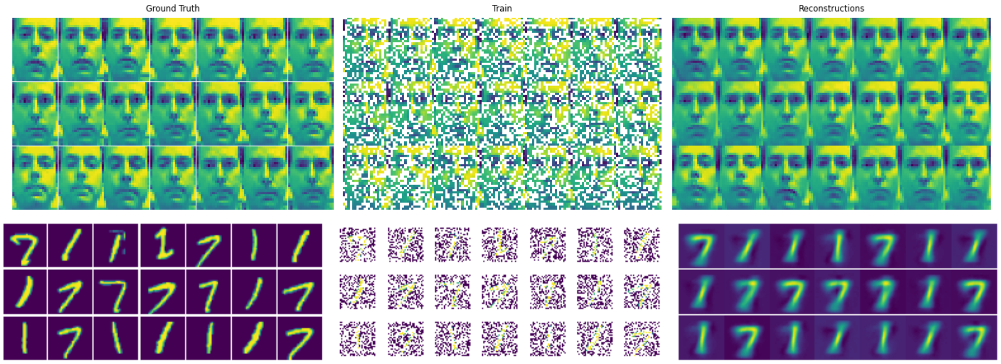


Figure 5: Top Row: Brendan faces reconstruction task with 39% missing pixels. Bottom row: MNIST reconstruction task where the digits were trained on partially observed images. In both rows the left column denotes ground truth data, the center column denotes a subset of the training data and the right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

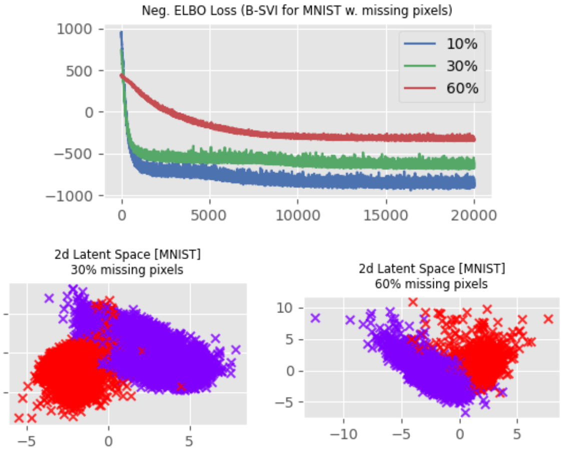


Figure 6: Top: ELBO loss for training with different degrees of missing pixels. Bottom: 2d latent space corresponding to the smallest lengthscales, note it is possible to disentangle the 2 digit classes quite effectively with almost no degradation for double the fraction of missing pixels.

generation from the learnt $5d$ latent distribution. Note that this reconstruction experiment differs from the less challenging *test-time* missing data which has been demonstrated in related work Titsias and Lawrence [2010]; Gal et al. [2014].

To demonstrate the versatility of the reconstruction task we tested the method on several examples of the *walking*, *jumping* and *running* 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 the B-SVI formulation. A sample reconstruction for a training point and a test point is shown in fig. 7.

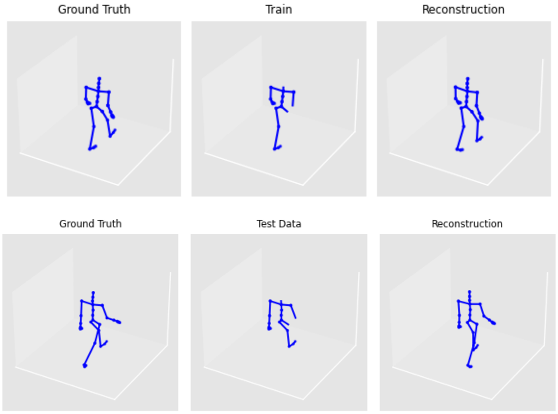


Figure 7: The train and test reconstruction of a single high-dimensional human pose (top: walking) and (bottom: running). The whole training exercise was conducted on incomplete silhouettes to extrapolate to sensible human poses at test time. For instance, the hand location during running was missing in this particular instance and was reconstructed to a remarkable similarity with the ground truth. We include several reconstructions in the supplementary.

4.3 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

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

Table 4: Test NLPD for datasets with ± standard error across 5 optimisation runs. The NLPDs across formulations indicate superior uncertainty quantification for the Bayesian schemes for both full and missing data problems.

Dataset	Test NLPD				Missing(%)
	Point	MAP	B-SVI (Ours)	AEB-SVI	
Oilflow	4.104 (3.223)	8.16 (1.224)	**-11.3105 (0.243)**	**-11.392 (0.147)**	–
qPCR	32.916 (3.39)	30.899 (2.399)	**27.844 (1.429)**	**25.422 (2.004)**	–
MOCAP	35472.566 (445.82)	8904.162 (162.45)	**2275.021 (33.89)**	–	44.5%

Table 5: Test RMSE for training with different degrees of missing dimensions per datapoint. The quality of reconstruction is best when the % missing during training matches the fraction of missing dimensions during testing.

Dataset	% missing (→ train %)		10%	30%	60%
MNIST	(↓ test %)	10%	0.2716	0.2735	0.2763
		30%	0.2731	0.2730	0.2794
		60%	0.2755	0.2762	0.2748

Table 6: The run time-comparisons highlight the important caveat that the amortised scheme (despite $\mathcal{O}(1)$ test-time predictions) is much slower (2x) as the encoder weights are global variational parameters which are updated at every mini-batch iteration as opposed to B-SVI.

Dataset	Avg. Iterations/sec.			
	Point	MAP	B-SVI (Ours)	AEB-SVI
Oilflow	167.56	168.42	164.43	89.42
qPCR	133.59	126.41	113.85	54.61
MOCAP	161.72	159.64	140.79	-

movie lens data and assess the quality of uncertainty estimates (for the reconstructed ratings) obtained with the B-SVI model (see fig. 8).

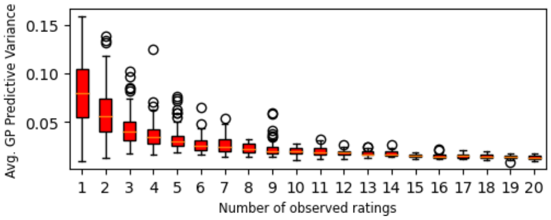


Figure 8: GP predictive variance around the ratings for a movie as a function of how many users have rated the movie. The higher the number of times a movie has been rated, the less uncertain is its prediction and vice-versa.

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.

VAEs: Deep probabilistic generative models like VAEs [Kingma and Welling, 2014] represent a related class of models where the decoder is a parameterised neural network. They have been hugely popular as an unsupervised learning tool for modelling images, large-scale object segmentation and frequently rely on convolutional neural nets as part of the encoding architecture. The most prominent variants include [Higgins et al., 2016], Kim and Mnih [2018], and [Sohn et al., 2015] which focus on disentanglement in latent space as a way to target superior output reconstruction. Structured VAEs need a large amount of input data to train and are unsuitable for tasks with only a moderate sized datasets (such as those used in the ablation study).

6 Conclusion

This paper introduces a generalised inference strategy for GPLVM models with key properties of scalable inference, flexible latent variable formulations, likelihoods and most importantly 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 [Kingma and Welling, 2014] and allows for robust, interpretable uncertainty around the predictions. 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. We showed in experiments that a fully Bayesian training procedure in conjunction with SVI yields excellent test time performance in these settings. The approach can be seamlessly extended to learn richer variational families in latent space along with missing data. Future work would focus in that direction.

Acknowledgements

This work was supported by the Wellcome Trust grant reference: 221052/Z/20/Z, 221052/A/20/Z, 221052/B/20/Z, 221052/C/20/Z, and 221052/E/20/Z. We thank Carl Henrik Ek and Eric Bodin for feedback on earlier versions of this manuscript. We also thank anonymous reviewers for their comments and feedback which helped clarify the message of the manuscript. VL acknowledges funding from the Qualcomm Innovation Fellowship (EU).

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.

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.

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *arXiv preprint arXiv:1809.11165*, 2018.

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.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

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.

H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

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.

S. Murray and H. Kjellström. Mixed likelihood gaussian process latent variable model. *arXiv preprint arXiv:1811.07627*, 2018.

nyc.gov. TLC Trip Record Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2020. [Online].

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.

S. Ramchandran, M. Koskinen, and H. Lähdesmäki. Latent gaussian process with composite likelihoods and numerical quadrature. In *International Conference on Artificial Intelligence and Statistics*, pages 3718–3726. PMLR, 2021.

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.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

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.

Supplementary Material: Generalised Gaussian Process Latent Variable Models (GPLVM) with Stochastic Variational Inference

A Broader Impact & Limitations

This work contributes a scalable method of inference for Bayesian GPLVM models used for non-parametric, probabilistic dimensionality reduction. Unsupervised learning tasks involving high-dimensional data are ubiquitous in the modern world. Some concrete examples are single-cell RNA data, financial time-series and medical records. 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]. One can identify structure in the high-dimensional data by analysing the clustering of low-dimensional latent factors. In the last few years there has been a proliferation of probabilistic generative models using deep neural networks like variational auto-encoders and variants which work extremely well on large and structured datasets, however canonical Bayesian GPLVM models [Titsias and Lawrence, 2010] originally worked best on small to moderate sized datasets. With the introduction of B-SVI in this work we further extend their domain to larger datasets. Further, the reason they adapt well to smaller datasets comes down to the non-parametric nature of Gaussian processes. Since these models concern non-parametric and probabilistic dimensionality reduction we believe these models can be useful in a much broader range of problems. Further, the fact that these models can train in the presence of missing data is a significant advantage and several real world datasets like medical records, corrupted images and ratings data are only partially observed. There is no straightforward way to deal with missing data in parametric models. Some important pitfalls to keep in mind when training with these models is the difficulty of assessing convergence and the variance of the doubly stochastic ELBO. It is important to ensure that the parameters of the latent distributions have converged, further one must carefully tune experimental parameters like the combination of batch-size and learning rate to achieve optimal performance.

B Relationship to [Murray and Kjellström, 2018]

Murray and Kjellström [2018] use the non-back constrained model with SVI along with non-Gaussian likelihoods but the scope of their experiments is limited to small datasets (max dimension 80) and assess the quality of clustering in the latent space comparing across likelihoods. They do experiment with missing values by dropping some attributes from 20% of the data, this means that the model can still sees full ground-truth on the remaining 80% of the data. This is very different to our framework where we conduct a systematic study of robustness of the model when training in the presence of massively missing data. We study the case when all training data is incomplete and has a high % of randomly missing attributes in each point, training and test (we show high fidelity reconstructions for MNIST and MOCAP which can be seen in further results section of the supplementary). An important point is that the testing/predictive framework summarised in section 3.4 (main paper) has not been effectively explored in more recent literature as the non-trivial setting requires a re-optimisation of the augmented ELBO to learn the latent points of the unseen $\mathbf{y}^*$. This is one of the reasons a lot of literature, for instance, Ramchandran et al. [2021]; Bui and Turner [2015] resort to the amortised set-up.

C Theory & Derivations

C.1 Motivation for inducing variables

The sparse inducing variable formulation is integral to the tractability of the Bayesian GPLVM. In order to see this, we proceed to derive a lower bound without inducing variables. As is standard, we wish to minimize the KL divergence between the variational approximation and the true posterior given by, $\mathrm{KL}(q(\{\boldsymbol{f}_d\}_{d=1}^D, \mathbf{X})||p(\{\boldsymbol{f}\}_{d=1}^D, \mathbf{X}|\mathbf{Y}))$.

Collecting all the $\boldsymbol{f}_d$'s in $\mathbf{F}$ for ease of notation.

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{F},\mathbf{X})||p(\mathbf{F},\mathbf{X}|\mathbf{Y})) &= \int q(\mathbf{F},\mathbf{X})\log\frac{q(\mathbf{F},\mathbf{X})}{p(\mathbf{F},\mathbf{X}|\mathbf{Y})}d\mathbf{F}d\mathbf{X} && (14)\\
&= -\underbrace{\int q(\mathbf{F},\mathbf{X})\log\frac{p(\mathbf{Y}|\mathbf{F},\mathbf{X})p(\mathbf{F}|\mathbf{X})p(\mathbf{X})}{q(\mathbf{F},\mathbf{X})}d\mathbf{F}d\mathbf{X}}_{\text{ELBO}} + \log p(\mathbf{Y}) && (15)
\end{aligned}$$

The evidence lower bound shown above is mathematically and computationally intractable due to the term $p(\mathbf{F}|\mathbf{X}) = \prod_{d=1}^{D} p(\boldsymbol{f}_d|\mathbf{X}) = \prod_{d=1}^{D}\mathcal{N}(\mathbf{0}, K_{nn}^{(d)})$ involving the variables $\mathbf{X}$ which appear non-linearly in the kernel matrix. The augmented bound constructed with inducing variables $\boldsymbol{u}_d$ for each dimension circumvents this intractability by leading to the cancellation of the difficult term in red.

C.2 Derivation of the DS-ELBO

We introduce auxiliary inducing variables, $\boldsymbol{u}_d \in \mathbb{R}^M$ for each of the latent functions $\boldsymbol{f}_d$. Variational inference in the augmented $(\mathbf{F},\mathbf{U},\mathbf{X})$ space is tractable.

The augmented variational approximation,

$$p(\mathbf{F},\mathbf{U},\mathbf{X}|\mathbf{Y}) \approx q(\mathbf{F},\mathbf{U},\mathbf{X}) = \prod_{d=1}^{D}[p(\boldsymbol{f}_d|\boldsymbol{u}_d,\mathbf{X})q(\boldsymbol{u}_d)]\prod_{n=1}^{N}q(\boldsymbol{x}_n) \qquad (16)$$

leads to the following KL between the approximation and the true posterior,

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{F},\mathbf{U},\mathbf{X})||p(\mathbf{F},\mathbf{U},\mathbf{X}|\mathbf{Y})) &= \int p(\mathbf{F}|\mathbf{U},\mathbf{X})q(\mathbf{U})q(\mathbf{X})\log\frac{p(\mathbf{F}|\mathbf{U},\mathbf{X})q(\mathbf{U})q(\mathbf{X})}{p(\mathbf{F},\mathbf{U},\mathbf{X}|\mathbf{Y})}d\mathbf{F}d\mathbf{U}d\mathbf{X}\\
&= -\int p(\mathbf{F}|\mathbf{U},\mathbf{X})q(\mathbf{U})q(\mathbf{X})\log\frac{p(\mathbf{Y}|\mathbf{F},\mathbf{X})\cancel{p(\mathbf{F}|\mathbf{U},\mathbf{X})}p(\mathbf{U}|Z)p(\mathbf{X})}{\cancel{p(\mathbf{F}|\mathbf{U},\mathbf{X})}q(\mathbf{U})q(\mathbf{X})}d\mathbf{F}d\mathbf{U}d\mathbf{X}\\
&\qquad + \log p(\mathbf{Y})
\end{aligned}$$

The final ELBO is given by,

$$\mathcal{L} = \int p(\mathbf{F}|\mathbf{U},\mathbf{X})q(\mathbf{U})q(\mathbf{X})\log\frac{p(\mathbf{Y}|\mathbf{F},\mathbf{X})p(\mathbf{U}|Z)p(\mathbf{X})}{q(\mathbf{U})q(\mathbf{X})}d\mathbf{F}d\mathbf{U}d\mathbf{X} \qquad (17)$$

C.3 Derivation of the expected likelihood term eq. (6)

In this section we explicitly tackle the triple integration in the expected likelihood term.

$$\begin{aligned}
\mathcal{L}_1 &= \sum_{n,d} \mathbb{E}_{p(\boldsymbol{f}_d|\boldsymbol{u}_d,\mathbf{X})q(\boldsymbol{u}_d)q(\boldsymbol{x}_n)}[\log p(y_{n,d}|\boldsymbol{f}_d,\boldsymbol{x}_n)] \qquad (18)\\
&= \sum_{n,d} \int q(\boldsymbol{x}_n) \int q(\boldsymbol{u}_d) \underbrace{\int p(\boldsymbol{f}_d|\boldsymbol{u}_d,\mathbf{X}) \log p(y_{n,d}|\boldsymbol{f}_d,\boldsymbol{x}_n) d\boldsymbol{f}_d}_{\mathcal{L}_f^{(n,d)}} d\boldsymbol{u}_d d\boldsymbol{x}_n\\
&= \sum_{n,d} \int q(\boldsymbol{x}_n) \underbrace{\int q(\boldsymbol{u}_d)\ \mathcal{L}_f^{(n,d)} d\boldsymbol{u}_d}_{\mathcal{L}_u^{(n,d)}} d\boldsymbol{x}_n\\
&= \sum_{n,d} \underbrace{\int q(\boldsymbol{x}_n)\ \mathcal{L}_u^{(n,d)} d\boldsymbol{x}_n}_{\mathcal{L}_{\mathbf{X}}^{(n,d)}} .
\end{aligned}$$

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

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

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 $p(\boldsymbol{f}_d|\boldsymbol{u}_d,\mathbf{X}) = \mathcal{N}(K_{nm}K_{mm}^{-1}\boldsymbol{u}_d, K_{nn} - K_{nm}K_{mm}^{-1}K_{mn})$. 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 we),

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

where $\Lambda_n = K_{mm}^{-1}k_n k_n^T K_{mm}^{-1}$ (Note: The $M \times M$ matrix $K_{mn}K_{nm}$ can be decomposed as $\sum_{n=1}^{N} k_n k_n^T$). Now, what remains is to perform the integration w.r.t $q(\boldsymbol{x}_n)$.

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

We note that the only terms involving the latent points $\boldsymbol{x}_n$ are K_{nm}, K_{nn} and $K_{nm}K_{mn}$; due to the summation at the beginning of the equation we can decompose the matrix terms into terms only dependent on the respective data point $\boldsymbol{x}_n$.

$$\begin{aligned}
= \sum_{n,d} \Big\{ \log \mathcal{N}(y_{n,d}| \underbrace{\langle k(\boldsymbol{x}_n, Z)\rangle_{q(\boldsymbol{x}_n)}}_{\Psi_1^{(n,\cdot)}} K_{mm}^{-1}\boldsymbol{m}_d, \sigma_y^2) - \frac{1}{2\sigma_y^2}\mathrm{Tr}(\underbrace{\langle k(\boldsymbol{x}_n,\boldsymbol{x}_n)\rangle_{q(\boldsymbol{x}_n)}}_{\psi_0^n}) \qquad (22)\\
+ \frac{1}{2\sigma_y^2}\mathrm{Tr}(K_{mm}^{-1} \underbrace{\langle k(Z,\boldsymbol{x}_n)k(\boldsymbol{x}_n,Z)\rangle_{q(\boldsymbol{x}_n)}}_{\Psi_2^n}) - \frac{1}{2\sigma_y^2}\mathrm{Tr}(S_d K_{mm}^{-1} \underbrace{\langle k(Z,\boldsymbol{x}_n)k(\boldsymbol{x}_n,Z)\rangle_{q(\boldsymbol{x}_n)}}_{\Psi_2^n} K_{mm}^{-1})\Big\}
\end{aligned}$$

where,

$$k(\boldsymbol{x}_n, Z) = k_n^T \ \ (n^{th} \text{ row of matrix } K_{nm}, \text{ dimension } 1 \times M) \tag{23}$$

$$k(Z, \boldsymbol{x}_n)k(\boldsymbol{x}_n, Z) = k_n k_n^T \ \ (\text{dimension } M \times M) \tag{24}$$

$$k(\boldsymbol{x}_n, \boldsymbol{x}_n) = K_{nn}^{(n,n)} \ \ (n^{th} \text{ entry on the diagonal of matrix } K_{nn} \text{ dimension } 1 \times 1) \tag{25}$$

C.4 Ψ statistics

In this section we show that expectations of the full covariance matrices K_{nm}, K_{nn} and $K_{nm}K_{mn}$ w.r.t $q(\mathbf{X})$ are indeed factorisable across data points.

$$\psi_0 = \text{Tr}(\langle K_{nn}\rangle_{q(\mathbf{X})}) = \Big\langle \sum_{n=1}^{N} K_{nn}^{(n,n)} \Big\rangle_{q(\mathbf{X})} = \sum_{n=1}^{N} \big\langle K_{nn}^{(n,n)} \big\rangle_{q(\boldsymbol{x}_n)}, \quad (q(\mathbf{X}) = \prod_{n=1}^{N} q(\boldsymbol{x}_n)) \tag{26}$$

$$= \sum_{n=1}^{N} \psi_0^n \tag{27}$$

Next, we look at Ψ_1,

$$\Psi_1 = \langle K_{nm} \rangle_{q(\mathbf{X})} \tag{28}$$

$$K_{nm} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{z}_M) \\ k(\boldsymbol{x}_2, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_2, \boldsymbol{z}_M) \\ \vdots & \vdots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{z}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{z}_M) \end{bmatrix} = \begin{bmatrix} \text{---} & K_{nm}^{(1,\cdot)} & \text{---} \\ \text{---} & K_{nm}^{(2,\cdot)} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & K_{nm}^{(N,\cdot)} & \text{---} \end{bmatrix} \tag{29}$$

$$\Psi_1 = \begin{bmatrix} \text{---} & \Psi_1^{(1,\cdot)} & \text{---} \\ \text{---} & \Psi_1^{(2,\cdot)} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \Psi_1^{(N,\cdot)} & \text{---} \end{bmatrix} = \begin{bmatrix} \text{---} & \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \text{---} \\ \text{---} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} & \text{---} \end{bmatrix}, \tag{30}$$

where we notice that Ψ_1 is a $N \times M$ matrix where each row just depends on a data point $\boldsymbol{x}_i$.

$$\Psi_2 = \langle K_{mn} K_{nm} \rangle_{q(\mathbf{X})} \tag{31}$$

$$= \begin{bmatrix} | & | & \vdots & | \\ \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \dots & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} \\ | & | & \vdots & | \end{bmatrix} \begin{bmatrix} \text{---} & \langle K_{nm}^{(1,\cdot)} \rangle_{q(\boldsymbol{x}_1)} & \text{---} \\ \text{---} & \langle K_{nm}^{(2,\cdot)} \rangle_{q(\boldsymbol{x}_2)} & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \langle K_{nm}^{(N,\cdot)} \rangle_{q(\boldsymbol{x}_N)} & \text{---} \end{bmatrix} \tag{32}$$

$$= \sum_{n=1}^{N} \langle K_{nm}^{(n,\cdot)^T} K_{nm}^{(n,\cdot)} \rangle_{q(\boldsymbol{x}_n)} \tag{33}$$

$$= \sum_{n=1}^{N} \Psi_2^n \tag{34}$$

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 $\boldsymbol{x}_i$.

C.5 KL divergence between factorised Gaussians

In eq. (5) in the main paper we re-write the KL term involving $q(\mathbf{X})$ as a factorisation across n, we show the proof below:

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

D Further Results

D.1 qPCR: Visualisation of Latent Space and relevance parameters

We analyse the qPCR dataset and recover the 10 cell developmental stages with our algorithm under each method, however, the point methods underperform the Bayesian methods in terms of disentanglement and also overfit when trained with 11 latent dimensions (see fig. 9). Both B-SVI and AEB-SVI give a clean recovery of cell developmental stages from the $48d$ data.

D.2 Oilflow: Automatic Relevance Determination

In this experiment we train the oilflow dataset with the same latent dimensions as data dimensions and learn the dominant dimensions from the kernel lengthscales. For the three models POINT, MAP and B-SVI the training errors were [0.0074, 0.0105, 0.0590] and test errors were [0.349, 0.527, 0.214] respectively. With more latent dimensions the point methods catastrophically overfit and fail to disentangle the three classes, while B-SVI manages to efficiently recover the dominant dimensions (see fig. 10).

D.3 NYC Taxi-cab: Test Reconstructions

In the plots in fig. 11 we visualise the ground-truth and predicted reconstructions per dimension (this corresponds to the 3 different taxi types operating in NYC namely - yellow, green and for-hire) over the whole test period of 10 days, each point indicates the total number of trips per hour.

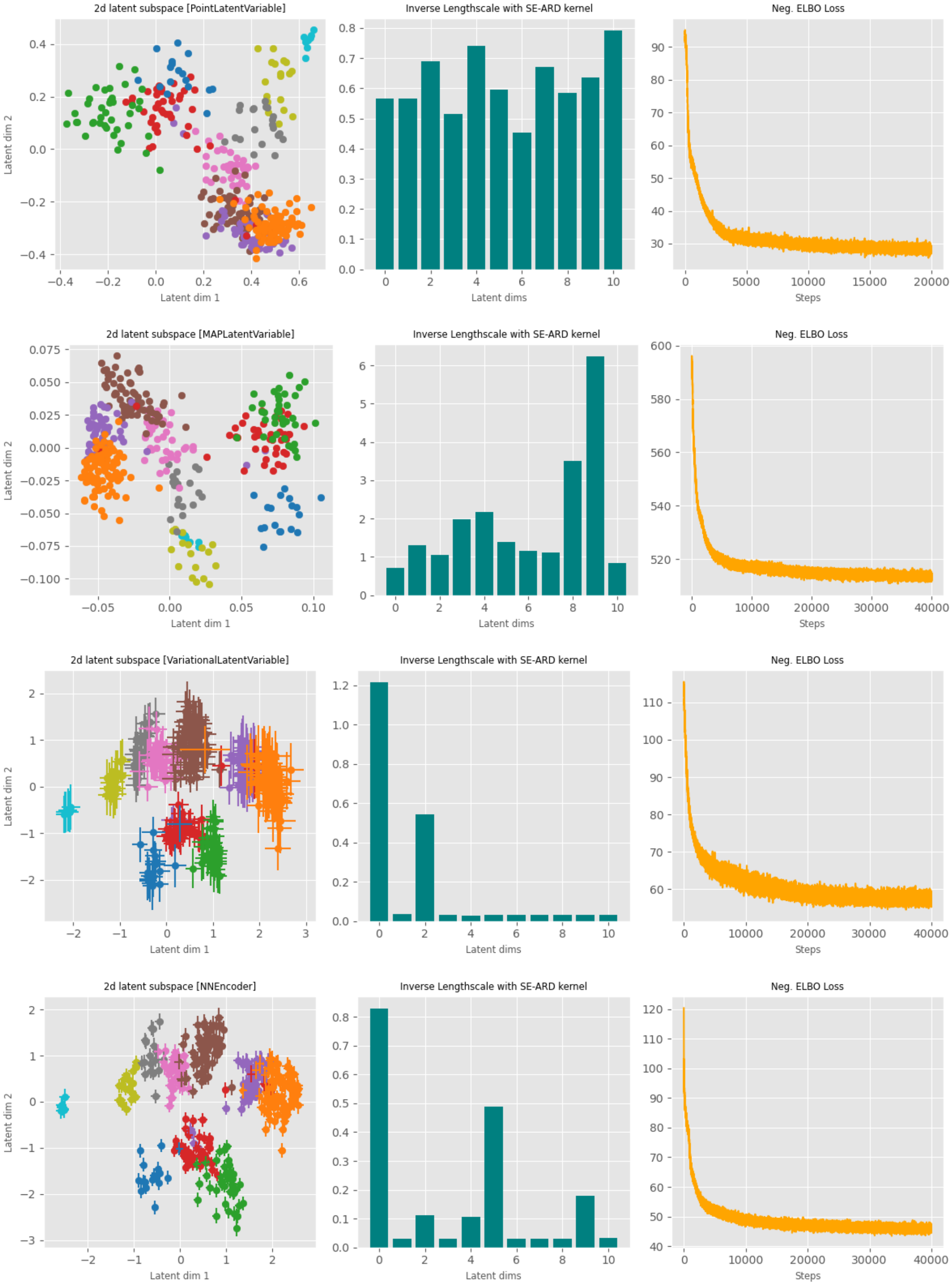


Figure 9: Analysis of qPCR data across models Point, MAP, B-SVI and AEB-SVI. For the bayesian models (bottom 2 rows) the vertical and horizontal lines crossing each point denote axis aligned Gaussian uncertainty of 1 standard deviation.

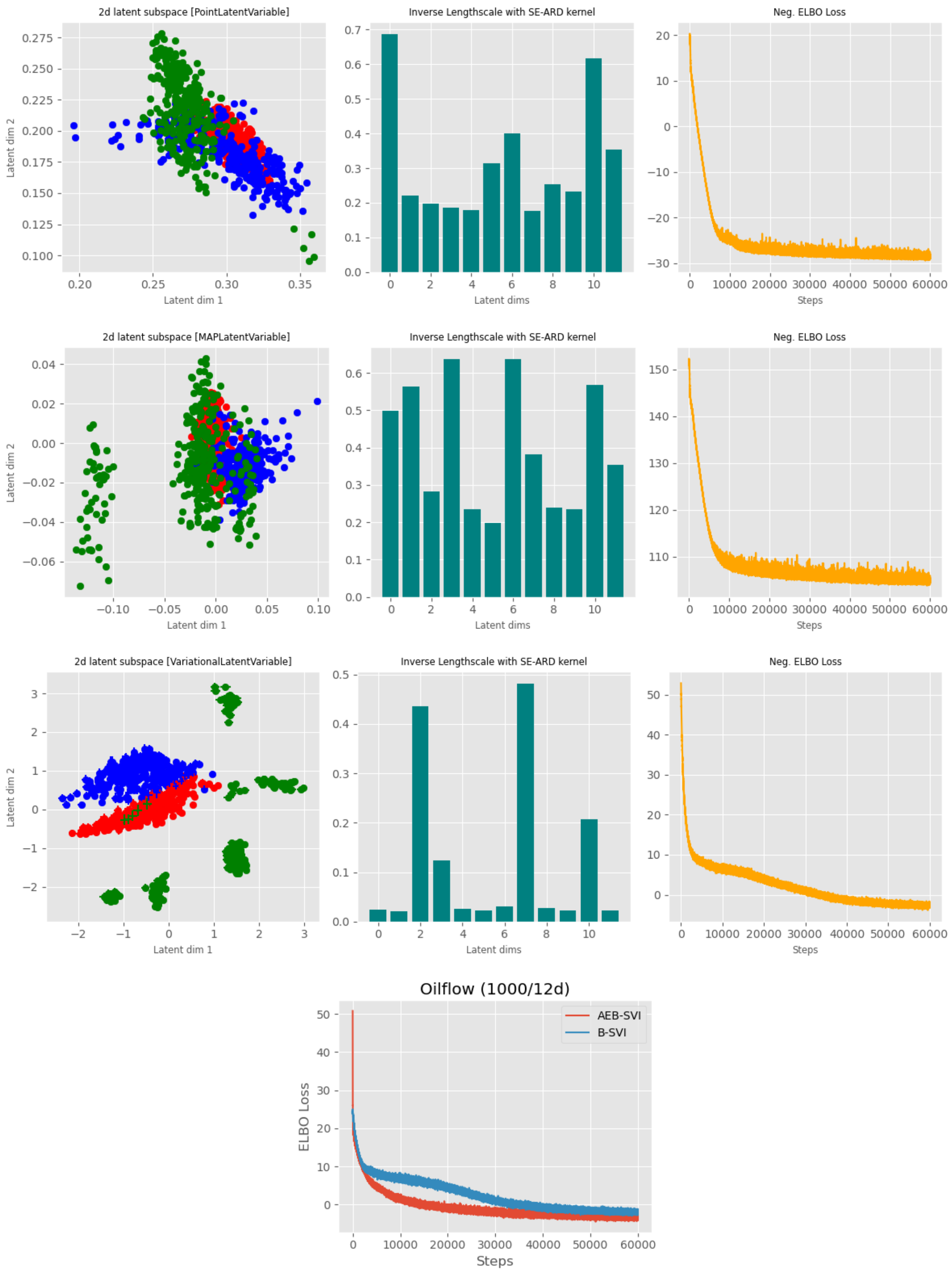
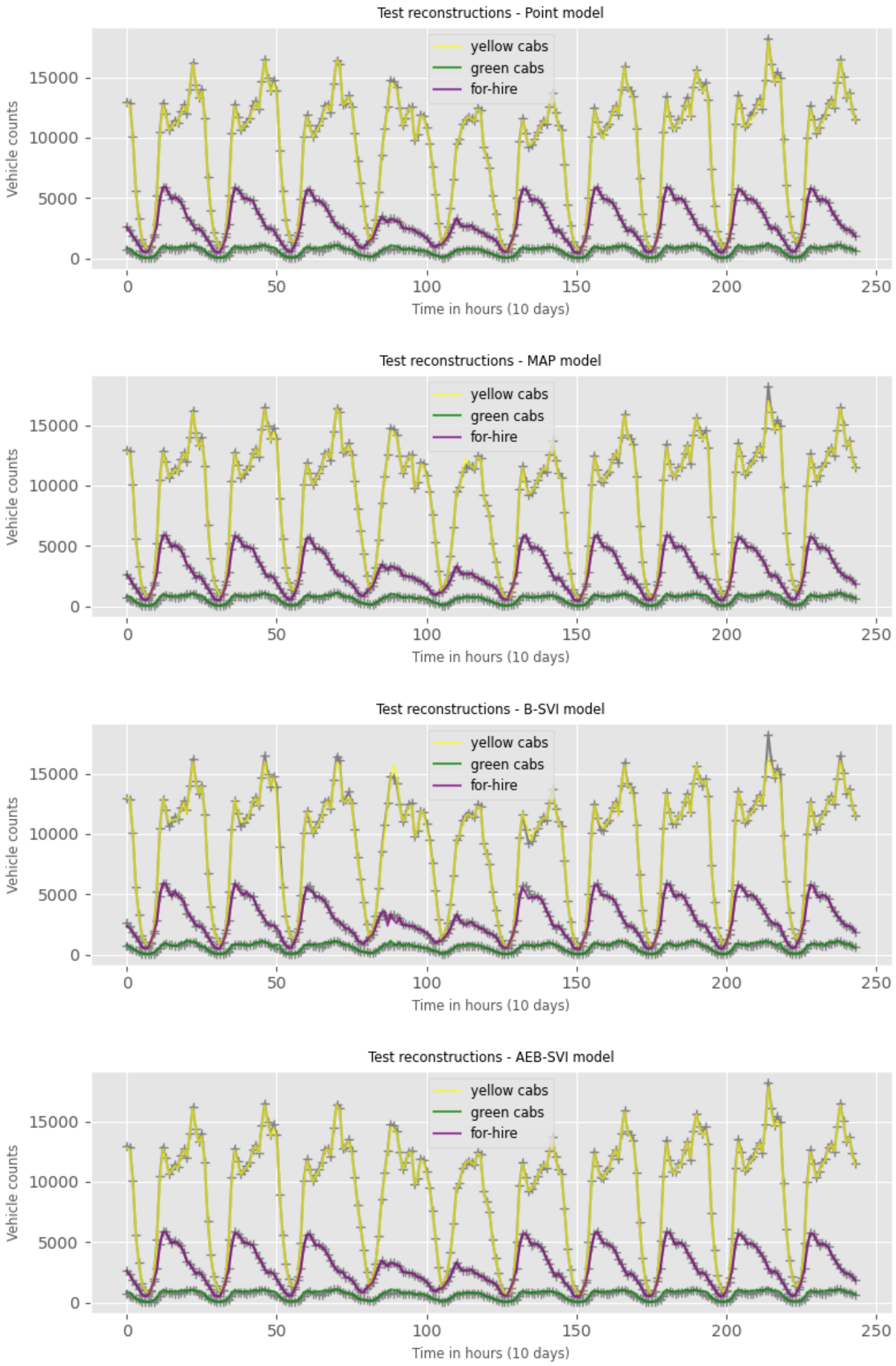


Figure 10: Analysis of oilflow dimensionality reduction with ARD. Final plot shows ELBO loss for B-SVI (non-amortised) and the amortised NNEncoder model with the latter achieving a very similar convergence loss level to the non-amortised model.



Test reconstructions - Point model

Test reconstructions - MAP model

Test reconstructions - B-SVI model

Test reconstructions - AEB-SVI model

Figure 11: Test reconstructions per taxi type over a 10 day period. All models are able to reconstruct the test time vehicle counts accurately, the spike on day 9 for yellow cabs is marginally underestimated for the Gaussian (B-SVI) and MAP model for this run, the error from that spike contributes ≈ 122 to the average RMSE. The test RMSE's for the above plots are: 97.12, 104.97, 176.51, 97.03 for the 4 models (in order of above) respectively.

D.4 MNIST: Missing data reconstructions

The plots below demonstrate reconstruction abilities when the algorithm is trained only on partially observed data. The missing pixels are distributed randomly for every image. Despite masking a large fraction of the pixels, the correct structure is reconstructed with only marginal degradation.

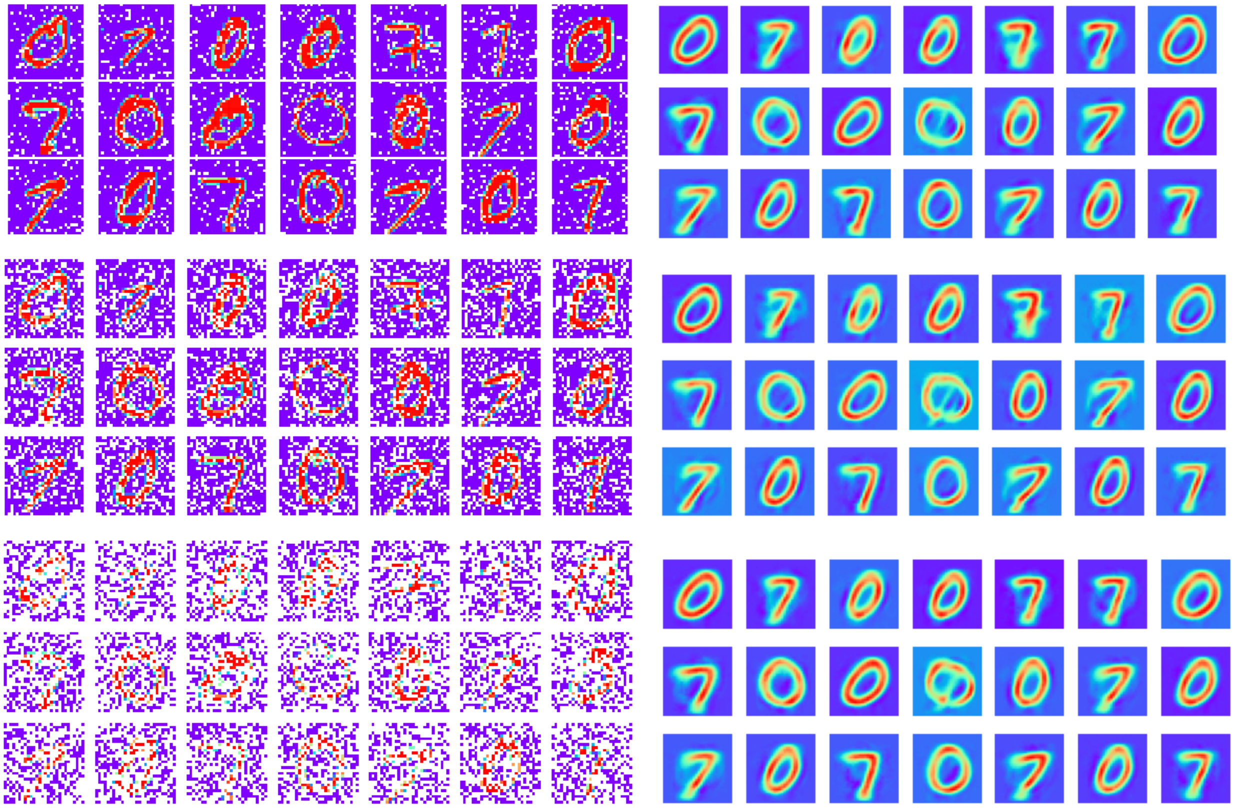


Figure 12: MNIST images reconstruction task for different degrees of missingness (10%, 30%, 60%). Left: Training digit. Right: Reconstruction post training.

D.5 MOCAP: Missing data reconstructions

We train B-SVI on 62d human motion capture data to try and recreate the sequence of diverse motions (walking, jumping and running) for a single subject. In order to test the models ability to learn in the presence of missing data we remove dimensions corresponding to different body parts to simulate different types of *missingness*. We cycle over the following types of structural missingness: (a) missing head, right leg and forearm, (b) missing forearms and left leg (c) missing upper body (d) missing lower body. Overall, the model yields very sensible reconstructions given the challenge of arbitrarily missing data. We note that the walking motion has the best reconstruction while jumping and running yield much superior reconstructions at training rather than test time. This is because the model had never seen the arm motion during running (arm data was missing from the training point) hence at test time the arm motion defaults to walking but the leg strides are captured accurately.

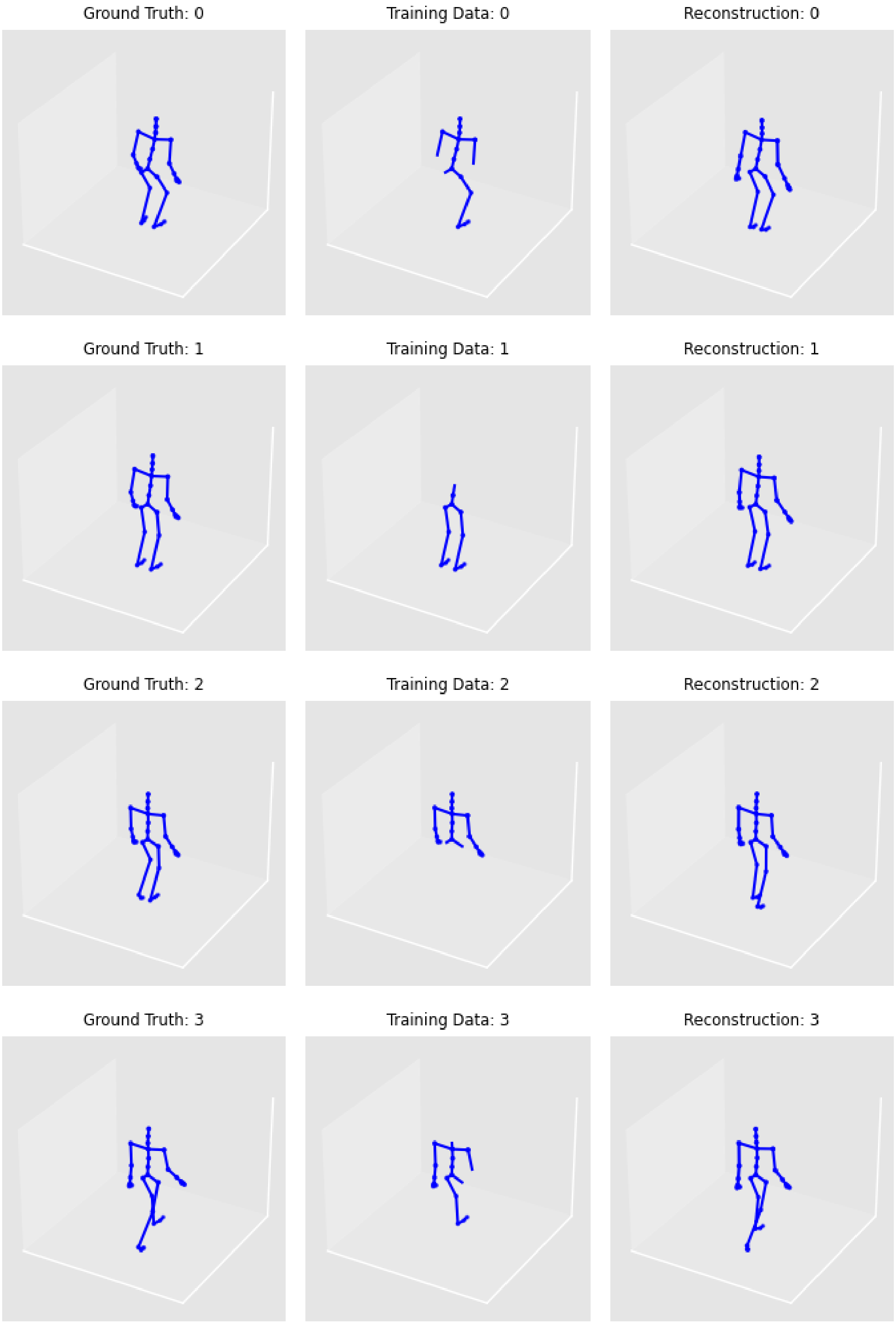


Figure 13: MOCAP reconstructions of missing dimensions on training data

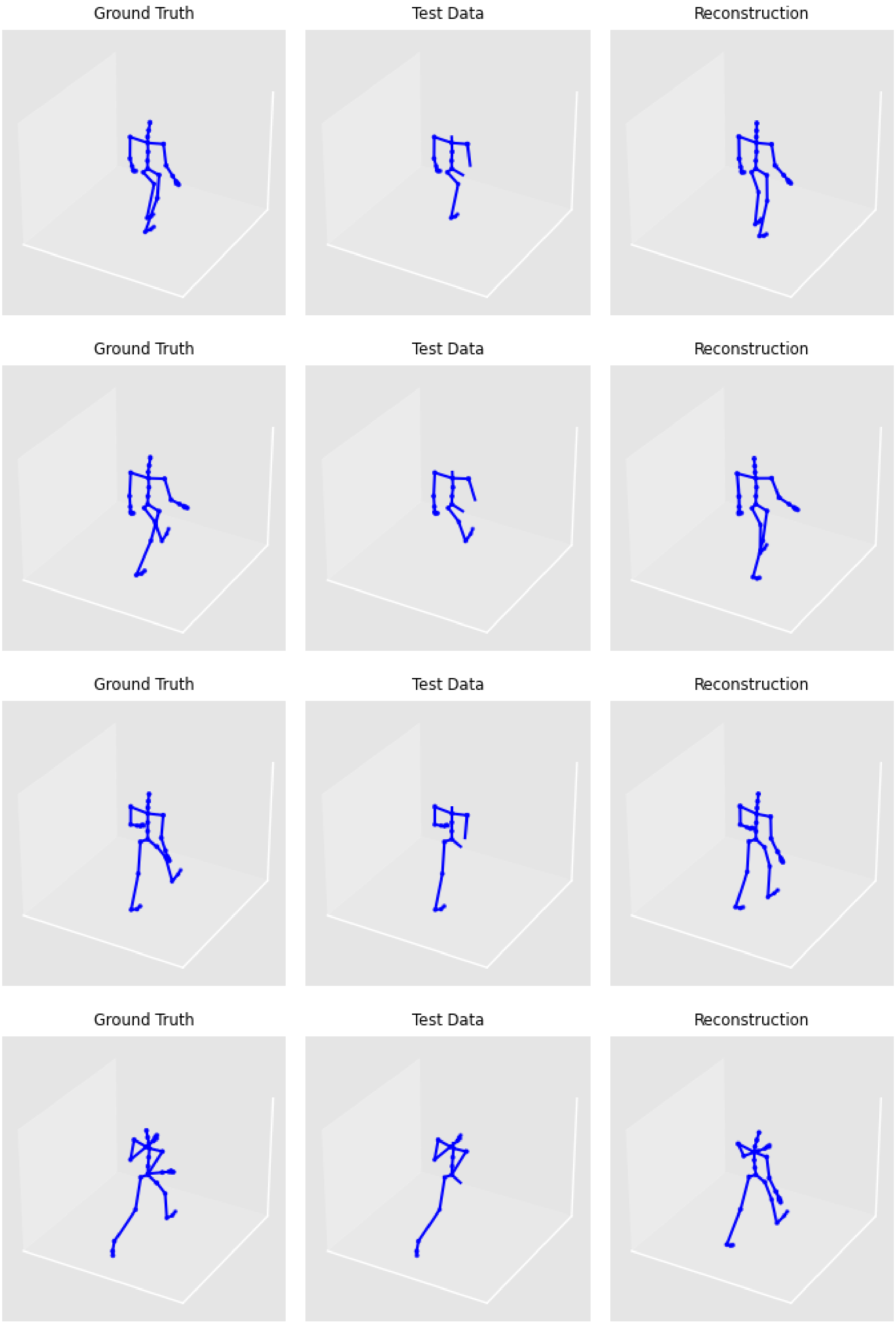


Figure 14: MOCAP reconstructions of missing dimensions on test data

D.6 Flexible Variational Families using Normalising Flows

In this section, we briefly introduce the use of normalising flows in our framework for capturing richer, non-Gaussian distributions in latent space.

Instead of parameterising $q(\mathbf{X})$ as a Gaussian we can parameterise it as a transformed Gaussian distribution by using a sequence of invertible and differentiable transformations. The variational distribution of B-SVI with a transformed Gaussian distribution is given by,

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

where the parameters of the flow mappings g_j are collected in ζ. We perform Monte Carlo expectations of the terms in the uncollapsed lower bound that involve $q_{\boldsymbol{\phi}}(\boldsymbol{x}_n)$ by sampling from the base Gaussian at each step $\boldsymbol{x}_n^{(0)} \sim \mathcal{N}(\mu_n, s_n \mathbb{I}_Q)$ and passing them through the flow $g_k \circ g_{k-1} \circ \ldots g_1(\boldsymbol{x}_n^{(0)})$ to yield the final latent point $\boldsymbol{x}_n^{(k)}$.

One can model each row of $\mathbf{X}$ (latent point) by an independent flow, although the number of parameters to be estimated in this model becomes too unwieldy very quickly. This approach also does not address the independence assumption across different data points. An alternative model that shares flow parameters across different latent points is tractable but is highly constrained to learn a flexible distribution per individual latent point and the trained flow-based distributions appear to be rather similar to a multivariate normal latent distribution only capturing local correlations but not non-linear correlations.

An interesting case is to use a single flow to model the joint density $q(\mathbf{X})$, so in the case of 2 latent points in $2d$ we learn a four dimensional flow based distribution modelling vec($\mathbf{X}$). Unsurprisingly, the distributions learnt in this model closely approximate the true posterior of the latent variables obtained using HMC (see fig. 15).

In this demo experiment, we generate a toy synthetic dataset using the forward model of the GPLVM (with a linear kernel, so this is equivalent to probabilistic PCA) of two points in $\mathbb{R}^{10}$, we attempt to learn a 2d latent space for this toy 2 point dataset using Algorithm 1. described in the paper except that $q(\mathbf{X})$ is non-Gaussian. We visualise samples from posterior distributions corresponding to the two $10d$ points in latent space.

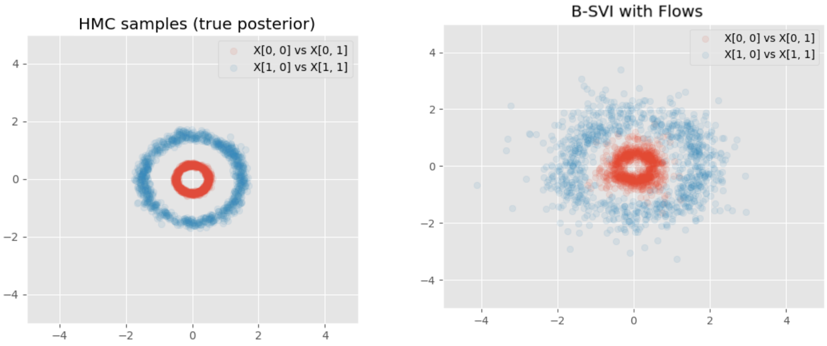


Figure 15: A demonstration of the effectiveness of the doubly stochastic algorithm to learn normalising flow based latent distributions for two individual high-dimensional points. *Left:* HMC samples from the true posterior. *Right:* Samples from the optimised flow based variational distribution with a base Gaussian and a sequence of 40 planar flows.

As PCA is rotation invariant, the true posterior of the latent variable is unidentifiable under rotations. This can be seen in the HMC samples drawn from the posterior of each latent point. We see that the SVI based model augmented with planar flows is able to capture this non-Gaussianity.

However, to achieve this result, many optimisation tricks were employed (e.g. increasing the number of samples to approximate the KL-divergence calculation, using a very small learning rate 1e-05 and long training times). The quality of results for flow based variational families for small toy-examples closely resemble gold-standard HMC but optimisation remains a difficulty and further research is required to understand how to achieve good training performance for moderate sized datasets.

E Experimental Configuration

Dataset	N	D	Z	Q	LR	Mini-batch	Train w. missing
Oilflow	1000	12	25	10	1e-03	100	No
qpCR	450	48	40	11	1e-03	100	No
Taxi-cab	744	3	36	2	5e-03	500	No
MNIST	15K	768	100	5	0.01	100	Yes
Brendan	1965	560	120	5	0.01	450	Yes
MOCAP	533	62	30	6	0.01	200	Yes
MovieLens	943	1682	34	15	0.005	100	Yes

Table 7: Training experimental configuration where N and D denote the number of data points and data space dimensions, Z denotes the number of inducing inputs shared across dimensions, Q denotes the dimesionality of the latent space, LR denotes the learning rate, β denotes the scalar annealing factor for the KL latent term in the ELBO.

F AEB-SVI: Network Architecture

We use two separate MLPs to encode the mean and covariance matrix, we use 2 hidden weight layers with tanh non-linearity. We summarise the network architecture (with the input and output layers) for learning the mean vector and covariance matrix below:

-Oilflow ($12D$)

a. Mean network: $(12(D), 10, 5, 12(Q))$ b. Covariance network: $(12(D), 78, 78, 144(Q^2))$

The number of nodes in the hidden layers for the covariance network are derived as $(D + Q^2)/2$

-qPCR ($48D$)

a. Mean network: $(48(D), 10, 5, 11(Q))$ b. Covariance network: $(48(D), 84, 84, 121(Q^2))$

-Taxi-cab ($3D$)

a. Mean network: $(3(D), 3, 3, 2(Q))$ b. Covariance network: $(3(D), 3, 3, 4(Q^2))$

Overall, we found that the latent representations learnt in the amortised case where not extremely sensitive to the architecture as long as sufficient capacity was reached. For qPCR, the latent dimensionality of $Q = 11$ was high enough to express the effective dimensionality of the data as the test reconstructions did not improve much for a higher latent dimensionality. Although, selecting $Q = 48$ for fully automatic model selection would not overfit in the Bayesian case, it would increase the compute time due the number of variational parameters and it is sensible to fine-tune Q to an appropriate size. Q was fixed across models to facilitate a comparison.

G Code Contribution

We wrote a custom implementation of Bayesian GPLVM in the `gpytorch` library [Gardner et al., 2018] to run the various latent variable configurations in this paper. We also wrote a custom Gaussian missing data likelihood class that can seamlessly handle NaNs in the $\mathbf{Y}$ data matrix. The code can be found attached to supplementary material and is publicly available.