Scalable Gaussian Process Variational Autoencoders

Metod Jazbec[1]
ETH Zürich

Matthew Ashman
University of Cambridge

Vincent Fortuin
ETH Zürich

Michael Pearce
University of Warwick

Stephan Mandt
University of California, Irvine

Gunnar Rätsch
ETH Zürich

Abstract

Conventional variational autoencoders fail in modeling correlations between data points due to their use of factorized priors. Amortized Gaussian process inference through GP-VAEs has led to significant improvements in this regard, but is still inhibited by the intrinsic complexity of exact GP inference. We improve the scalability of these methods through principled sparse inference approaches. We propose a new scalable GP-VAE model that outperforms existing approaches in terms of runtime and memory footprint, is easy to implement, and allows for joint end-to-end optimization of all components.

1 Introduction

Variational autoencoders (VAEs) are among the most widely used models in representation learning and generative modeling (Kingma and Welling, 2013, 2019; Rezende et al., 2014). As VAEs typically use factorized priors, they fall short when modeling correlations between different data points. However, more expressive priors that capture correlations enable useful applications. Casale et al. (2018), for instance, showed that by modeling prior correlations between the data, one could generate a digit's rotated image based on rotations of the same digit at different angles.

Gaussian process VAEs (GP-VAEs) have been designed to overcome this shortcoming (Casale et al., 2018). These models introduce a Gaussian process (GP) prior over the latent variables that correlates the latent variables through a kernel function. While GP-VAEs have outperformed standard VAEs on many tasks (Casale et al., 2018; Fortuin et al., 2020; Pearce, 2020), combining the GPs and VAEs brings along fundamental computational challenges. On the one hand, neural networks reveal their full power in conjunction with large datasets, making mini-batching a practical necessity. GPs, on the other hand, are traditionally restricted to medium-scale datasets due to their unfavorable scaling. In GP-VAEs, these contradictory demands must be reconciled, preferably by reducing the $\mathcal{O}(N^3)$ complexity of GP inference, where N is the number of data points.

Despite recent attempts to improve the scalability of GP-VAE models by using specifically designed kernels and inference methods (Casale et al., 2018; Fortuin et al., 2020), a generic way to scale these models, regardless of data type or kernel choice, has remained elusive. This limits current GP-VAE implementations to small-scale datasets. In this work, we introduce the first generically scalable method for training GP-VAEs based on inducing points. We thereby improve the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(bm^2 + m^3)$, where m is the number of inducing points and b is the batch size.

We show that applying the well-known inducing point approaches (Hensman et al., 2013; Titsias, 2009) to GP-VAEs is a non-trivial task: existing sparse GP approaches cannot be used off-the-shelf within GP-VAE models as they either necessitate having the entire dataset in the memory or do not lend themselves to being amortized. To address this issue, we propose a simple hybrid sparse GP method that is amenable to both mini-batching and amortization.

We make the following contributions:

- We propose the first scalable GP-VAE framework

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

[1]Contact: jazbec.metod@gmail.com

based on sparse GP inference (Sec. 3). In contrast to existing methods, our model is agnostic to the kernel choice, makes no assumption on the structure of the data at hand and allows for joint optimization of all model components.

- We provide theoretical motivations for the proposed method and introduce a hybrid sparse GP model that accommodates a crucial demand of GP-VAEs for simultaneous amortization and batching.
- We show empirically that the proposed approximation scheme maintains a high accuracy while being much more scalable and efficient (Sec. 4). Importantly from a practitioner's point of view, our model is easy to implement as it requires no special modification of the training procedure.

2 Related Work

Sparse Gaussian processes. There has been a long line of work on sparse Gaussian process approximations, dating back to Snelson and Ghahramani (2006), Quiñonero-Candela and Rasmussen (2005), and others. Most of these sparse methods rely on a summarizing set of points referred to as *inducing points* and mainly differ in the exact way of selecting those. Variational learning of inducing points was first considered in Titsias (2009) and was shown to lead to significant performance gains. Instead of optimizing an approximate marginal GP likelihood as done in non-variational sparse models, a lower bound on the exact GP marginal likelihood is derived and used as a training objective. Another approach relevant for our work is the stochastic variational approach from Hensman et al. (2013), where the authors proposed a sparse model that can, in addition to reducing the GP complexity, also be trained in mini-batches, enabling the use of GP models on (extremely) large datasets.

Improving VAEs. Extending the expressiveness and representational power of VAEs can be roughly divided into two (orthogonal) approaches. The first one focuses on increasing the flexibility of the approximate posterior (Rezende and Mohamed, 2015; Kingma et al., 2016), while the second one consists of imposing a richer prior distribution on the latent space. Various extensions to the standard Gaussian prior have been proposed, including a Gaussian mixture prior (Dilokthanakul et al., 2016; Kopf et al., 2019), hierarchical structured priors (Johnson et al., 2016; Deng et al., 2017), and a von Mises-Fisher distribution prior (Davidson et al., 2018). GP-VAE models are part of this second group and, contrary to other work on extending VAE priors, aim to relax the *iid* assumption between data points. Moreover, GP-VAEs are also related to approaches that aim to learn more structured and interpretable representations of the data by incorporating auxiliary information, such as time or viewpoints (Sohn et al., 2015; Lin et al., 2018; Johnson et al., 2016).

Gaussian process VAEs. As mentioned above, the most related approaches to our work are the GP-VAE models of Casale et al. (2018) and Pearce (2020). However, neither of these are scalable for generic kernel choices and data types. The model from Pearce (2020) relies on exact GP inference, while Casale et al. (2018) exploit a (partially) linear structure of their GP kernel and use a Taylor approximation of the ELBO to get around computational challenges. Another GP-VAE model is proposed in Fortuin et al. (2020) where it is used for multivariate time series imputation. Their model is indeed scalable (even in linear time complexity), but it works exclusively on time series data since it exploits the Markov assumption. Additionally, it does not support a joint optimization of GP parameters, but assumes a fixed GP kernel.

3 Scalable SVGP-VAE

This work's main contribution is the sparsification of the GP-VAE using the sparse GP approaches mentioned above. To this end, two separate variational approximation problems have to be solved jointly: an outer amortized inference procedure from the high-dimensional space to the latent space, and the inner sparse variational inference scheme on the GP. To motivate our proposed solution, we begin by pointing out the problems that arise when naïvely combining the two objectives.

3.1 Problem setting and notation

In this work, we consider high-dimensional data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times K}$. Each data point has a corresponding low-dimensional auxiliary data entry, summarized as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathcal{X}^N, \mathcal{X} \subseteq \mathbb{R}^D$. For example, $\mathbf{y}_i$ could be a video frame and $\mathbf{x}_i$ the corresponding time stamp. Our goal is to train a model for (1) generating $\mathbf{Y}$ conditioned on $\mathbf{X}$ and (2) infering an interpretable and disentangled low-dimensional representations.

To this end, we adopt a latent GP approach, summarized below. First, we need to model a prior distribution over the collection of latent variables $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^T \in \mathbb{R}^{N \times L}$, each latent variable $\mathbf{z}_i$ living in an L-dimensional latent space. To model their joint distribution, we assume L independent latent functions $f^1, \ldots, f^L \sim GP(0,\, k_\theta)$ with kernel parameters

θ that result in $\mathbf{Z}$ when being evaluated on $\mathbf{X}$. More precisely, $\mathbf{z}_i = [f^1(\mathbf{x}_i), \dots, f^L(\mathbf{x}_i)]$. By construction, the l^{th} latent channel of all latent variables $\mathbf{z}^l_{1:N} \in \mathbb{R}^N$ (the l^{th} column of $\mathbf{Z}$) has a correlated Gaussian prior with covariance $\mathbf{K}_{NN} = k_\theta(\mathbf{X}, \mathbf{X})$. Setting $\mathbf{K}_{NN} = I$ recovers the fully factorized prior commonly used in standard VAEs.

As in regular VAEs, each $\mathbf{z}_i \in \mathbb{R}^L$ is then "decoded" to parameterize the distribution over observations $\mathbf{y}_i = \mu_\psi(\mathbf{z}_i) + \boldsymbol{\varepsilon}_i$ where $\mu_\psi : \mathbb{R}^L \to \mathbb{R}^K$ is a network with parameters ψ and $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(\mathbf{0},\, \sigma_y^2\, \mathbf{I}_K)$. Mathematically, the full generative model is given by

$$p_\theta(\mathbf{Z}|\mathbf{X}) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{z}^l_{1:N}|0, \mathbf{K}_{NN}),$$
$$p_\psi(\mathbf{Y}|\mathbf{Z}) = \prod_{i=1}^{N} p_\psi(\mathbf{y}_i|\mathbf{z}_i) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i|\mu_\psi(\mathbf{z}_i), \sigma_y^2\, \mathbf{I}_K).$$

The joint distribution is $p_{\psi,\theta}(\mathbf{Y}, \mathbf{Z}|\mathbf{X}) = p_\psi(\mathbf{Y}|\mathbf{Z}) p_\theta(\mathbf{Z}|\mathbf{X})$. The true posterior for the latent variables $p_{\psi,\theta}(\mathbf{Z}|\mathbf{Y}, \mathbf{X}) = p_{\psi,\theta}(\mathbf{Y}, \mathbf{Z}|\mathbf{X}) / p_{\psi,\theta}(\mathbf{Y}|\mathbf{X})$ is intractable due to the denominator which requires integrating over $\mathbf{Z}$. Hence, approximate inference methods are required to infer the unobserved $\mathbf{Z}$ given the observed $\mathbf{X}$ and $\mathbf{Y}$.

3.2 Amortized variational inference

Amortization in the typical VAE architecture uses a second (inference) network from the high-dimensional data $\mathbf{y}_i$ to the mean and variance of a fully factorized Gaussian distribution over $\mathbf{z}_i \in \mathbb{R}^L$ (Zhang et al., 2018). We denote it as $\tilde{q}_\phi(\mathbf{z}_i|\mathbf{y}_i) = \mathcal{N}(\mathbf{z}_i|\mu_\phi(\mathbf{y}_i), \mathrm{diag}(\sigma^2_\phi(\mathbf{y}_i)))$ and it has network parameters ϕ. In Casale et al. (2018), this Gaussian distribution is used directly to approximate the posterior, $p_{\psi,\theta}(\mathbf{Z}|\mathbf{Y}) \approx \prod_i \tilde{q}_\phi(\mathbf{z}_i|\mathbf{y}_i)$. While this approach mirrors classical VAE design, the approximate posterior for a latent variable $\mathbf{z}_i$ only depends on $\mathbf{y}_i$ and ignores $\mathbf{x}_i$. This is in stark contrast to traditional Gaussian processes where latent function values $f(x)$ are informed by all y values according to the similarity of the corresponding x values.

Building on this model, Pearce (2020) instead proposed to use the inference network $\tilde{q}_\phi(\mathbf{z}_i|\mathbf{y}_i)$ to replace only the intractable likelihood $p_\psi(\mathbf{y}_i|\mathbf{z}_i)$ in the posterior. By combining $\tilde{q}_\phi$ with tractable terms, the approximate posterior could be explicitly normalized as

$$q(\mathbf{Z}|\mathbf{Y}, \mathbf{X}, \phi, \theta) := \prod_{l=1}^{L} \frac{\prod_{i=1}^{N} \tilde{q}_\phi(\mathbf{z}^l_i|\mathbf{y}_i)\, p_\theta(\mathbf{z}^l_{1:N}|\mathbf{X})}{Z^l_{\phi,\theta}(\mathbf{Y}, \mathbf{X})}, \quad (1)$$

where the normalizing constant $Z^l_{\phi,\theta}(\mathbf{Y}, \mathbf{X})$ can be computed analytically. Noting the symmetry of the Gaussian distribution, $\mathcal{N}(z|\mu, \sigma) = \mathcal{N}(\mu|z, \sigma)$, the approximate posterior for channel l is mathematically equivalent to the (exact) GP posterior in the traditional GP regression with inputs $\mathbf{X}$ and outputs $\tilde{\mathbf{y}}_l := \mu^l_\phi(\mathbf{Y})$ with heteroscedastic noise $\tilde{\boldsymbol{\sigma}}_l := \sigma^l_\phi(\mathbf{Y})$. We therefore refer to each $\{\mathbf{X}, \tilde{\mathbf{y}}_l, \tilde{\boldsymbol{\sigma}}_l\}$ as the *latent dataset* for the l^{th} channel. Each normalizing constant of Equation 1 is also the GP marginal likelihood of the l^{th} latent dataset. The parameters $\{\psi, \phi, \theta\}$ are learnt by maximizing the evidence lower bound (ELBO) in the **P**earce model,

$$\mathcal{L}_P(\psi, \phi, \theta) = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{z}_i|\cdot)} \Big[\log p_\psi(\mathbf{y}_i|\mathbf{z}_i) - \log \tilde{q}_\phi(\mathbf{z}_i|\mathbf{y}_i) \Big] + \sum_{l=1}^{L} \log Z^l_{\phi,\theta}(\mathbf{Y}, \mathbf{X}). \quad (2)$$

The first term is the difference between the true likelihood and inference network approximate likelihood, while the second term is the sum over GP marginal likelihoods of each latent dataset.

One subtle, yet important, characteristic of the variational approximation from Pearce (2020) is that it gives rise to the ELBO $\mathcal{L}_P(\cdot)$ that contains the GP posterior. Note that this is in contrast to Casale et al. (2018) and Fortuin et al. (2020), where the GP prior is part of the ELBO. As we will show in Section 3.3, the ELBO that contains the GP posterior naturally lends itself to "sparsification" through the use of sparse GP posterior approximations.

The computational challenges of $\mathcal{L}_P(\cdot)$ are twofold. Firstly, for the latent GP regression, an inverse and a log-determinant of the kernel matrix $\mathbf{K}_{NN} \in \mathbb{R}^{N \times N}$ must be computed, resulting in $\mathcal{O}(N^3)$ time complexity. Secondly, the ELBO does not decompose as a sum over data points, so the entire dataset $\{\mathbf{X}, \mathbf{Y}\}$ is needed for one evaluation of $\mathcal{L}_P(\cdot)$.

Given the latent dataset, at first glance, we may simply apply sparse GP regression techniques instead of traditional regression. We next look at two widely used methods (Titsias (2009) and Hensman et al. (2013)) and highlight their drawbacks for this task. We then propose a new hybrid approach solving these issues.

3.3 Latent Sparse GP Regression

To simplify the notation, we focus on a single channel and suppress l, resulting in $\tilde{\mathbf{y}}$ and $\tilde{\boldsymbol{\sigma}}$, $\log Z_{\theta,\phi}(\cdot)$ and f. Given an (amortized latent) regression dataset $\mathbf{X}$, $\tilde{\mathbf{y}}$, $\tilde{\boldsymbol{\sigma}}$, sparse Gaussian process methods assume that there exists a set of $m \ll N$ inducing points with inputs $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathcal{X}^m$ and outputs $\mathbf{f}_m := f(\mathbf{U}) \sim \mathcal{N}(f(\mathbf{U})|\, \boldsymbol{\mu}, \mathbf{A})$ that summarize the regression

dataset. $\mathbf{U}$, $\boldsymbol{\mu}$, $\mathbf{A}$ are parameters to be learnt. Given a (test) set of r new inputs $\mathbf{X}_r$, the sparse approximate (predictive) distribution over outputs $\mathbf{f}_r = f(\mathbf{X}_r)$ is

$$q_S(\mathbf{f}_r|\mathbf{X}_r, \mathbf{U}, \boldsymbol{\mu}, \mathbf{A}, \theta) = \mathcal{N}\big(\mathbf{f}_r|\mathbf{K}_{rm}\mathbf{K}_{mm}^{-1}\boldsymbol{\mu},\ \mathbf{K}_{rr} - \mathbf{K}_{rm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mr} + \mathbf{K}_{rm}\mathbf{K}_{mm}^{-1}\mathbf{A}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mr}\big), \quad (3)$$

where kernel matrices are $\mathbf{K}_{mm} = k_\theta(\mathbf{U}, \mathbf{U})$, $\mathbf{K}_{rr} = k_\theta(\mathbf{X}_r, \mathbf{X}_r)$, and $\mathbf{K}_{mr} = \mathbf{K}_{rm}^\top = k_\theta(\mathbf{U}, \mathbf{X}_r)$. By introducing inducing points, the cost of learning the model is reduced from $\mathcal{O}(N^3)$ in $\log Z_{\phi,\theta}(\cdot)$ to $\mathcal{O}(Nm^2)$ in a modified objective.

We next describe two of the most popular ways to learn the variational parameters $\mathbf{U}$, $\boldsymbol{\mu}$, $\mathbf{A}$ that are based on a second *inner* variational approximation for the Gaussian process regression that lower bounds $\log Z_{\phi,\theta}(\cdot)$. For this second inner variational inference, we aim to learn a cheap $q_S(\cdot)$ (Equation 3) that closely approximates the expensive $q(\cdot)$ (Equation 1).

Titsias (2009). Let $\mathbf{z} = \mathbf{z}_{1:N}^l$, then the parameters $\mathbf{U}, \boldsymbol{\mu}, \mathbf{A}$ may be learnt by minimizing $\mathrm{KL}\big(q_S(\mathbf{z}\,|\cdot)\ ||\ q(\mathbf{z}\,|\cdot)\big)$, or equivalently by maximizing a lower bound to the marginal likelihood of the latent dataset $\log Z_{\phi,\theta}^l(\cdot)$. Let $\boldsymbol{\Sigma} := \mathbf{K}_{mm} + \mathbf{K}_{mN}\mathrm{diag}(\tilde{\boldsymbol{\sigma}}^{-2})\mathbf{K}_{Nm}$, then the optimal $\boldsymbol{\mu}$ and $\mathbf{A}$ may be found analytically:

$$\boldsymbol{\mu}_T = \mathbf{K}_{mm}\boldsymbol{\Sigma}^{-1}\mathbf{K}_{mN}\mathrm{diag}(\tilde{\boldsymbol{\sigma}}^{-2})\tilde{\mathbf{y}}, \quad (4)$$

$$\mathbf{A}_T = \mathbf{K}_{mm}\boldsymbol{\Sigma}^{-1}\mathbf{K}_{mm}, \quad (5)$$

where $\mathbf{K}_{mN} = k_\theta(\mathbf{U}, \mathbf{X})$. Plugging $\boldsymbol{\mu}_T$ and $\mathbf{A}_T$ back into the appropriate evidence lower bound yields the final lower bound for learning $\mathbf{U}$ in the **T**itsias model

$$\mathcal{L}_T(\mathbf{U}, \phi, \theta) = \log\mathcal{N}\big(\tilde{\mathbf{y}}|\mathbf{0},\ \mathbf{K}_{Nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mN} + \mathrm{diag}(\tilde{\boldsymbol{\sigma}}^2)\big) - \frac{1}{2}Tr\big(\mathrm{diag}(\tilde{\boldsymbol{\sigma}}^{-2})\,(\mathbf{K}_{NN} - \mathbf{K}_{Nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mN})\big). \quad (6)$$

Note that the bound is a function of $\tilde{\mathbf{y}}$ and $\tilde{\boldsymbol{\sigma}}$ which depend on the inference network with parameters ϕ and the kernel matrices which depend upon θ hence we make these arguments explicit. In the full GP-VAE ELBO $\mathcal{L}_P(\cdot)$, substituting $q_S(\cdot)$, $\mathcal{L}_T(\cdot)$ in place of $q(\cdot)$, $\log Z_{\phi,\theta}(\cdot)$ yields a sparse GP-VAE ELBO that can be readily used to reduce computational complexity of existing GP-VAE methods for a generic dataset and an arbitrary GP kernel function.[1]

However, observe from Equations 4, 5 and 6 that the entire dataset $\{\mathbf{X}, \mathbf{Y}\}$ enters through $\mathbf{K}_{NN}$ and $\tilde{\mathbf{y}}$, $\tilde{\boldsymbol{\sigma}}$ respectively. Therefore, this ELBO is not amenable to mini-batching and has large memory requirements.

Hensman et al. (2013). In order to make variational sparse GP regression amenable to mini-batching, Hensman et al. (2013) proposed an ELBO that lower bounds $\mathcal{L}_T$ and, more importantly, decomposes as a sum of terms over data points. Adopting our notation with explicit parameters, the **H**ensman ELBO is given by

$$\mathcal{L}_H(\mathbf{U}, \boldsymbol{\mu}, \mathbf{A}, \phi, \theta) = -\mathrm{KL}\big(q_S(\mathbf{f}_m|\cdot)\ ||\ p_\theta(\mathbf{f}_m|\cdot)\big) + \sum_{i=1}^{N}\Big\{\log\mathcal{N}\big(\tilde{y}_i|\boldsymbol{k}_i\mathbf{K}_{mm}^{-1}\boldsymbol{\mu},\ \tilde{\sigma}_i^{-2}\big) - \frac{1}{2\tilde{\sigma}_i^2}\,(\tilde{k}_{ii} + Tr(\mathbf{A}\,\Lambda_i))\Big\}. \quad (7)$$

Above, $\boldsymbol{k}_i$ is the i-th row of $\mathbf{K}_{Nm}$, $\Lambda_i = \mathbf{K}_{mm}^{-1}\boldsymbol{k}_i\boldsymbol{k}_i^\top\mathbf{K}_{mm}^{-1}$ and $\tilde{k}_{ii}$ is the i-th diagonal element of the matrix $\mathbf{K}_{NN} - \mathbf{K}_{Nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mN}$. Due to the decomposition over data points, the gradients $\nabla\mathcal{L}_H(\cdot)$ in stochastic or mini-batch gradient descent are unbiased and only the data in the current batch are needed in memory for the gradient updates. Consequently, with batch size b the GP complexity is further reduced to $\mathcal{O}(bm^2 + m^3)$. Note that for $\boldsymbol{\mu} = \boldsymbol{\mu}_T, \mathbf{A} = \mathbf{A}_T$ and $b = N$, $\mathcal{L}_H(\cdot)$ recovers $\mathcal{L}_T(\cdot)$ (Hensman et al., 2013).

While this method may seem to meet our requirements, it has a fatal drawback. Firstly, it is not amortized as $\boldsymbol{\mu}$ and $\mathbf{A}$ are not functions of the observed data $\{\mathbf{X}, \mathbf{Y}\}$ but instead need to be optimized once for each dataset. Secondly, as a consequence, in the full GP-VAE ELBO $\mathcal{L}_P(\cdot)$, substituting $q_S(\cdot)$, $\mathcal{L}_H(\cdot)$ in place of $q(\cdot)$, $\log Z_{\phi,\theta}(\cdot)$ and simplifying yields the following expression

$$\mathcal{L}_{PH}(\mathbf{U}, \psi, \theta, \boldsymbol{\mu}^{1:L}, \mathbf{A}^{1:L}) = \sum_{i=1}^{N}\mathbb{E}_{q_S}\Big[\log p_\psi(\mathbf{y}_i|\mathbf{z}_i)\Big] - \sum_{l=1}^{L}KL\big(q_S^l(\mathbf{f}_m|\cdot)\ ||\ p_\theta^l(\mathbf{f}_m|\cdot)\big) \quad (8)$$

where $q_S^l(\mathbf{f}_m|\cdot) = \mathcal{N}(\mathbf{f}_m|\boldsymbol{\mu}^l, \mathbf{A}^l)$.

Note that the ELBO above is not a function of the inference network parameters ϕ (for the full derivation, we refer to Appendix B.1). The sparse approximate posterior is parameterized by $\mathbf{U}, \boldsymbol{\mu}, \mathbf{A}, \theta$ which are all treated as free parameters to be optimized, that is, they are not functions of the latent dataset or the inference network. Maximizing the full GP-VAE ELBO is equivalent to minimizing the KL divergence from the approximate to the true posterior and neither of these depend upon the latent dataset or the inference network. Therefore, using the Hensman sparse GP within an amortized GP-VAE model causes the ELBO

[1] As an aside, this sparse GP-VAE ELBO may also be derived in the standard way using $\mathrm{KL}\big(q_S(\mathbf{Z}|\cdot)||p_{\psi,\theta}(\mathbf{Z}|\mathbf{Y}, \mathbf{X})\big)$, see Appendix B.4.

to be independent of the inference network parameters. Hence, this method also cannot be used as-is to amortize the sparse GP-VAE with mini-batches.

3.4 The best of both ELBOs

Recall our goal to make GP-VAE models amenable to large datasets. This requires avoiding the large memory requirements and being able to amortize inference. To alleviate these problems, Casale et al. (2018) propose to use a Taylor approximation of the GP prior term in their ELBO. However, this significantly increases implementation complexity and gives rise to potential risks in ignoring curvature. We take a different approach utilising sparse GPs. We desire a model that can scale to large datasets, like Hensman et al. (2013), while also being able to directly compute variational parameters from the latent regression dataset, like Titsias (2009). To this end, we take a mini-batch of the data, $\mathbf{X}_b \subset \mathbf{X}$, $\mathbf{Y}_b \subset \mathbf{Y}$, and with the network $\tilde{q}_\phi(\cdot)$ create a mini-batch of the latent dataset $\mathbf{X}_b$, $\tilde{\mathbf{y}}_b$, $\tilde{\boldsymbol{\sigma}}_b$. Following Titsias (2009), with Equations 4 and 5 for the optimal $\boldsymbol{\mu}_T$ and $\mathbf{A}_T$, we analytically compute stochastic estimates for each latent channel l given by

$$\begin{aligned}
\boldsymbol{\Sigma}_b^l &:= \mathbf{K}_{mm} + \frac{N}{b}\mathbf{K}_{mb}\,\mathrm{diag}(\tilde{\boldsymbol{\sigma}}_b^{-2})\,\mathbf{K}_{bm}, \\
\boldsymbol{\mu}_b^l &:= \frac{N}{b}\mathbf{K}_{mm}\left(\boldsymbol{\Sigma}_b^l\right)^{-1}\mathbf{K}_{mb}\,\mathrm{diag}(\tilde{\boldsymbol{\sigma}}_b^{-2})\,\tilde{\mathbf{y}}_b^l, \\
\mathbf{A}_b^l &:= \mathbf{K}_{mm}\left(\boldsymbol{\Sigma}_b^l\right)^{-1}\mathbf{K}_{mm}. \qquad (9)
\end{aligned}$$

where $\mathbf{K}_{mb} = k_\theta(\mathbf{U}, \mathbf{X}_b) \in \mathbb{R}^{m\times b}$. For a full derivation of these estimators, see Appendix B.2. All these estimators are consistent, so they converge to the true values for $b \to N$. However, while $\boldsymbol{\Sigma}_b^l$ is an unbiased estimator for $\boldsymbol{\Sigma}^l$, the same does not hold for $\boldsymbol{\mu}_b^l$ and $\mathbf{A}_b^l$. We investigate the magnitude of the bias in Appendix C.4 finding that it is generally small in practice. We believe this result to be in line with sparse Gaussian process approximations that assume the whole dataset may be summarized by a set of inducing points. Alternatively, this may be interpreted as assuming that the dataset contains redundancy, that is, that we have more than enough data to learn the latent function. In such a case, (cheaply) learning an average of latent functions of multiple mini-batches would closely approximate (expensively) learning one latent function using the full dataset.

$\boldsymbol{\mu}_b^l$ and $\mathbf{A}_b^l$ parameterize the approximate posterior $q_S(\cdot)$ which is, therefore, a direct function of the data $\mathbf{X}_b$, $\mathbf{Y}_b$ and hence it is an amortized approximate posterior. By taking a mini-batch of data, one may assume that we may also compute $\mathcal{L}_T(\cdot)$ of the mini-batch latent dataset. However, note that such an $\mathcal{L}_T(\cdot)$ is a lower bound for $\log Z_{\phi,\theta}(\cdot)$ of the mini-batch latent dataset, not a lower bound for the full latent dataset. Instead, we use $\boldsymbol{\mu}_b^l$ and $\mathbf{A}_b^l$ along with $\mathbf{U}$ and θ to compute the GP evidence lower bound of Hensman et al. (2013) given in Equation 7, which is also suitable to mini-batching and lower bounds the marginal likelihood of the full latent dataset. Finally, the evidence lower bound of our **S**parse (**V**ariational) **G**aussian **P**rocess **V**ariational **A**uto**e**ncoder, for a single mini-batch $\mathbf{X}_b, \mathbf{Y}_b$, is thus

$$\begin{aligned}
&\mathcal{L}_{SVGP-VAE}(\mathbf{U}, \psi, \phi, \theta) := \\
&\quad \sum_{i=1}^{b} \mathbb{E}_{q_S}\Big[\log p_\psi(\mathbf{y}_i|\mathbf{z}_i) - \log \tilde{q}_\phi(\mathbf{z}_i|\mathbf{y}_i)\Big] \\
&\qquad + \frac{b}{N}\sum_{l=1}^{L}\mathcal{L}_H^l(\mathbf{U},\, \phi,\, \theta,\, \boldsymbol{\mu}_b^l,\, \mathbf{A}_b^l), \qquad (10)
\end{aligned}$$

where each $\mathcal{L}_H^l(\cdot)$ is computed using the mini-batch of the latent dataset $\mathbf{X}_b$, $\tilde{\mathbf{y}}_b^l$, $\tilde{\boldsymbol{\sigma}}_b^l$. By naturally combining well known approaches, we arrive at a sparse GP-VAE that is both amortized and can be trained using mini-batches. The VAE parameters ϕ, ψ, inducing points $\mathbf{U}$, and the GP kernel θ can all be optimized jointly in an end-to-end fashion as we show in the next section.

Also note that during training, $\boldsymbol{\mu}_b^1, ..., \boldsymbol{\mu}_b^L$ and $\mathbf{A}_b^1, ..., \mathbf{A}_b^L$ are computed from a mini-batch $\mathbf{X}_b$, $\mathbf{Y}_b$. However at test time, given a new dataset, all available data $\mathbf{X}$, $\mathbf{Y}$ may be used to compute the $\boldsymbol{\mu}^1, .., \boldsymbol{\mu}^L$ and $\mathbf{A}^1, ..., \mathbf{A}^L$. The Gaussian process structure places no theoretical restriction upon the number of observations that are incorporated into the approximate posterior parameters, any amount of data can be pooled simply according to the kernel operations. In contrast, neural networks typically assume fixed input and output sizes and pooling data in a principled way requires much more attention.

While we have treated the auxiliary data $\mathbf{X}$ as observed throughout this section, our model can also be used when $\mathbf{X}$ is not given (or is only partly observed). In such cases, we make use of the Gaussian Process Latent Variable Model (GP-LVM) introduced by Lawrence (2004) to learn the missing part of $\mathbf{X}$, similar to what is done in Casale et al. (2018). In SVGP-VAE, (missing parts of) $\mathbf{X}$ can be learned jointly with the rest of the model parameters.

4 Experiments

We compared our proposed model with existing approaches measuring both performance and scalability on some simple synthetic data and large high-dimensional benchmark datasets. Implementation details can be found in Appendix A and additional experiments in Appendix C. The implementation of our

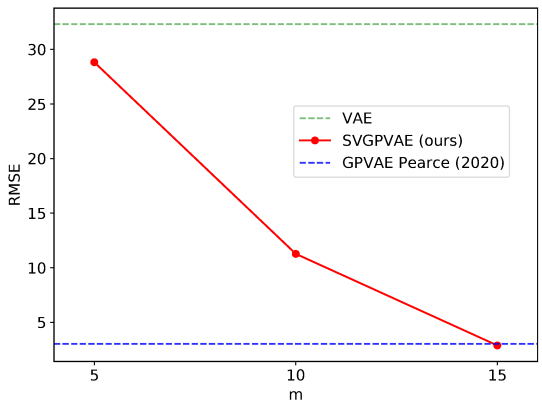

Figure 1: Performance of our SVGP-VAE models as a function of the number of inducing points. We see that as we increase the number of inducing points, the performance gracefully approaches the one of the exact GP-VAE baseline model.

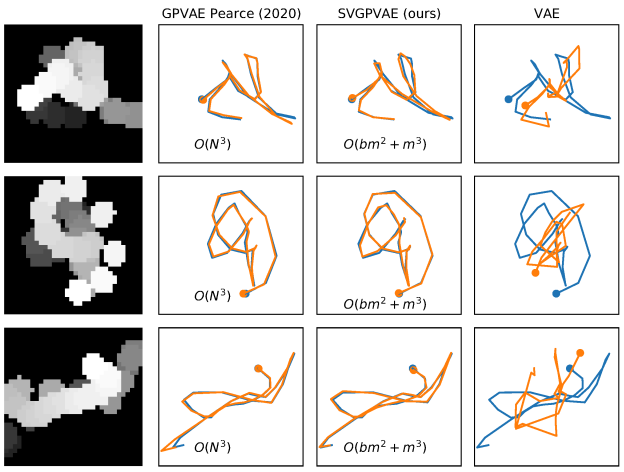

Figure 2: Reconstructions of the latent trajectories for the moving ball data. Frames of each test video are overlaid and shaded by time in the first column. Ground truth trajectories are depicted in blue, while predicted trajectories are shown in orange. We can see that the standard VAE fails to model the trajectories faithfully, while the GP-VAE models (including our sparse approximation) match them closely. Note that $b = N$ in SVGP-VAE for this experiment. For SVGP-VAE, the number of inducing points was set to $m = 15$.

model as well as our experiments are publicly available at `https://github.com/ratschlab/SVGP-VAE`.

4.1 Synthetic moving ball data

The moving ball data was utilized in Pearce (2020). It consists of black-and-white videos of a moving circle, where the 2D trajectory is sampled from a GP with radial basis function (RBF) kernel. The goal is to reconstruct the correct underlying trajectory in the two-dimensional latent space from the frames in pixel space. Since the videos are short (30 frames), full GP inference is still feasible in this setting, such that we can compare our sparse approach against the gold standard. Note that due to the small dataset size we do not perform mini-batching within each video here.

Scaling behavior. We see in Figure 1 that as we increase the number of inducing points our method uses, its performance in terms of root mean squared error (RMSE) approaches the performance of the full GP baselines. It reaches the baseline performance already with 15 inducing points, which is half the number of data points in the trajectory and therefore four times less computationally intensive than the baseline. The reconstructions of the trajectories also qualitatively agree with the baseline, as can be seen in Figure 2.

Optimization of kernel parameters. Another advantage of our proposed method over the previous approaches is that it is agnostic to the kernel choice and even allows to optimize the kernel parameters (and thereby learn a better kernel) jointly during training. In Pearce (2020), joint optimization of kernel parameters was not considered, while in Casale et al. (2018) a special training regime is deployed where VAE and GP parameters are optimized at different stages. Since the moving ball data is generated by a GP, we know the optimal kernel length scale for the RBF kernel in this case, which is namely the one of the generating process. We optimized the length scale of our SVGP-VAE kernel and found that when using a sufficient number of inducing points, we indeed recover the true length scale almost perfectly (Fig. 3). Note that when too few inducing points are used, the *effective* length scale of the observed process in the subspace spanned by these inducing points is indeed larger, since some of the variation in the data will be orthogonal to that subspace. It is thus to be expected that our model would also choose a larger length scale to model the observations in this subspace.

Optimization of inducing points. When working with sparse Gaussian processes, the selection of inducing point locations can often be crucial for the quality of the approximation (Titsias, 2009; Fortuin et al., 2018; Jähnichen et al., 2018; Burt et al., 2019). In our model, we can optimize these inducing point locations jointly with the other components. On the moving ball data, since the trajectories are generated from stationary GPs, the optimal inducing point loca-

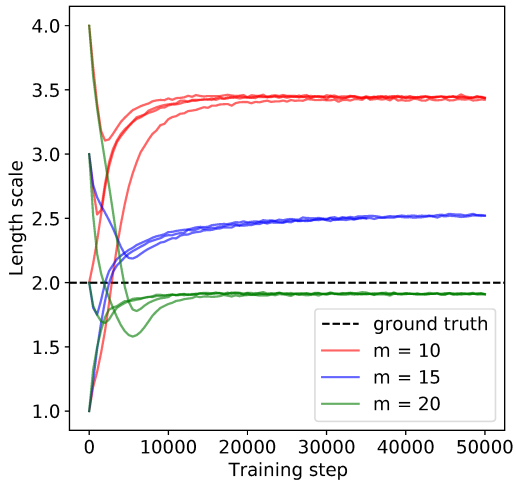

Figure 3: Optimized length scales of our SVGP-VAE model during training on the moving ball data. With sufficiently many inducing points, the model recovers the true length scale of the generating process.

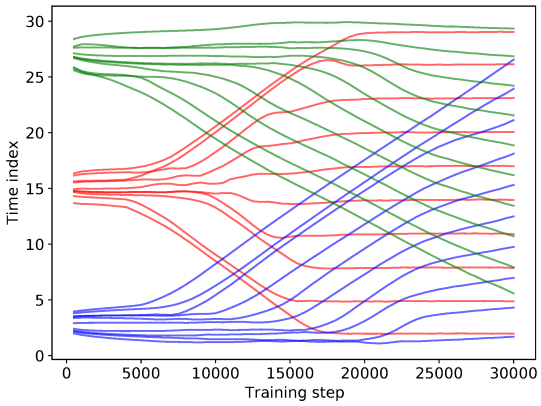

Figure 4: Optimized inducing points of our SVGP-VAE model during training on the moving ball data for three different (suboptimal) initializations. We can see that the model correctly learns to spread the inducing points evenly over the time series, which should be expected as a stationary GP kernel is used in the data generating process.

tions should be roughly equally spaced along the time dimension. When we adversarially initialize the inducing points in a small region of the time series, we see that the model pushes them apart over the course of training and converges to this optimal spacing (Fig. 4). Together with the previous experiment, these observations suggest that the model is able to choose close-to-optimal inducing points and kernel functions in a data-driven way during the normal training process.

4.2 Conditional generation of rotated MNIST digits

To benchmark our model against existing scalable GP-VAE approaches, we follow the experimental setup from Casale et al. (2018) and use rotated MNIST digits (LeCun et al., 1998) in a conditional generation task. The task is to condition on a number of digits that have been rotated at different angles and to generate an image of one of these digits rotated at an unseen angle. In the original work, they consider 400 images of the digit 3, each rotated at multiple angles in $[0, 2\pi]$. Using identical architectures, kernel, and dataset ($N = 4050$), we report results for both the GP-VAE of Casale et al. (2018) and our SVGP-VAE. The full GP-VAE model from Pearce (2020) cannot be applied to this size of data, hence it is omitted. As alternative baselines, we report results for a conditional VAE (CVAE) (Sohn et al., 2015) as well as for an extension of a sparse GP (SVIGP) approach from Hensman et al. (2013). We use the GECO algorithm (Rezende and Viola, 2018) to train our SVGP-VAE model, which greatly improves the stability of the training procedure.

Performance of conditional generation. We see in Table 1 that our proposed model outperforms the VAE baselines in terms of MSE, while still being computationally more efficient than the model from Casale et al. (2018) (in theory and practice).[2] This can also be seen visually in Figure 5 as our model produces the most faithful generations. For the SVGP-VAE, the number of inducing points was set to $m = 32$ and the batch size was set to $b = 256$. For the GP-VAE (Casale et al., 2018), the low-rank matrix factor H depends on the dimension of the linear kernel M used in their model ($M = 8$ and $H = 128$).

Moreover, our SVGP-VAE model comes close in performance to the unamortized sparse GP model with deep likelihood from Hensman et al. (2013). This shows that the amortization gap of our model is small (Cremer et al., 2018). Note that this baseline was not considered in the previous GP-VAE literature (Casale et al., 2018), even though for the task of conditional generation, where we try to learn a single GP over the entire dataset, amortization is not strictly needed. However, in tasks where the inference has to be amor-

[2]Note that in their paper, Casale et al. (2018) report a performance of 0.028 on this task. However, their code for the MNIST experiment is not openly available and we could not reproduce this result with our reimplementation (which is also available at `https://github.com/ratschlab/SVGP-VAE`).

Table 1: Results on the rotated MNIST digit 3 dataset. Reported here are mean values together with standard deviations based on 5 runs. We see that our proposed model performs comparably to the sparse GP baseline from Hensman et al. (2013) and outperforms the VAE baselines while still being more scalable than the Casale et al. (2018) model.

	MSE	**GP complexity**	**Time/epoch [s]**
CVAE (Sohn et al., 2015)	0.0796 ± 0.0023	-	0.39 ± 0.01
GPPVAE (Casale et al., 2018)	0.0370 ± 0.0012	$\mathcal{O}(NH^2)$	19.10 ± 0.66
SVGP-VAE (ours)	0.0251 ± 0.0005	$\mathcal{O}(bm^2 + m^3)$	1.90 ± 0.02
Deep SVIGP (Hensman et al., 2013)	0.0233 ± 0.0014	$\mathcal{O}(bm^2 + m^3)$	1.15 ± 0.04

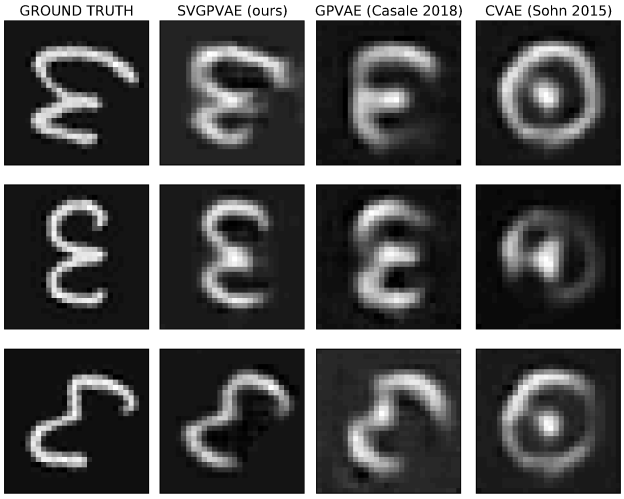

Figure 5: Conditionally generated rotated MNIST images. The generations of our proposed model are qualitatively more faithful to the ground truth. For more examples see Appendix C.3.

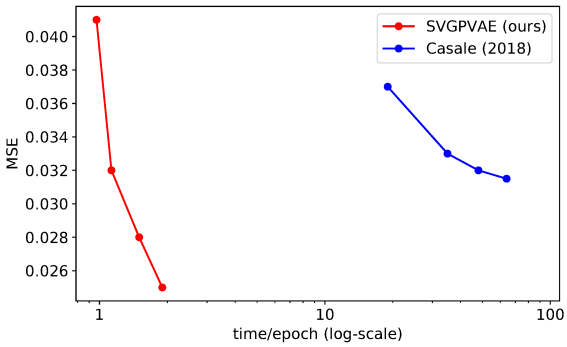

Figure 6: Performance of our proposed model with different numbers of inducing points and the Casale et al. (2018) model with different kernel dimensionalities as a function of runtime. For the SVGP-VAE, we consider four different configurations of inducing points, while for the Casale et al. (2018) model, we use four different dimensions of the linear kernel: $m, M \in \{8, 16, 24, 32\}$.

tized across several GPs, this model could not be used. More details on this baseline are provided in Appendix C.5.

Tradeoff between runtime and performance. The performance of our sparse approximation can be increased by choosing a larger number of inducing points, at a quadratic cost in terms of runtime. The Casale et al. (2018) model, while being more restricted in its kernel choice, offers a similar tradeoff between runtime and performance by choosing a different dimensionality for the low-rank linear kernel used in their latent space (see Appendix B.3). In Figure 6 we depict performance for both models when varying the number of inducing points and the dimension of the linear kernel, respectively. We observe that SVGP-VAE, besides being much faster, exhibits a steeper decline in the MSE as the model's capacity is increased.

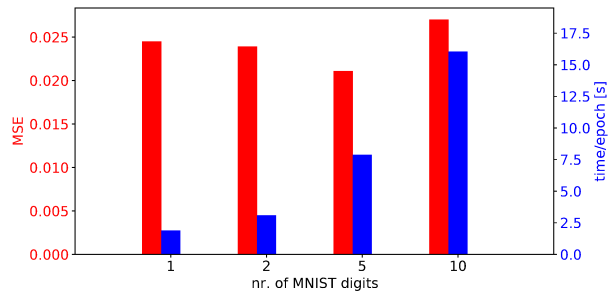

Figure 7: Performance and runtime of our proposed model on differently sized subsets of the MNIST dataset, including the full set. We see that the performance stays roughly the same, regardless of dataset size, while the runtime grows linearly as expected. The size of each dataset equals $4050 \times$ nr. of MNIST digits.

Scaling to larger data. As mentioned above, Casale et al. (2018) restrict their experiment to a small subset of the MNIST dataset and indeed we did also not manage to scale their model to the whole dataset on our hardware (11 GB GPU memory). Our SVGP-VAE, however, is easily scalable to such dataset sizes. We report its performance on larger subsets of MNIST (including the full dataset) in Figure 7. We see that the performance of our proposed model does not deteriorate with increased dataset size, while the runtime grows linearly as expected. All in all, we thus see that our model is more flexible than the previous GP-VAE approaches, scales to larger datasets, and achieves a better performance at lower computational cost.

4.3 SPRITES experiment

We additionally assessed the performance of our model on the SPRITES dataset (Li and Mandt, 2018). It consists of images of cartoon characters in different actions/poses. Each character has a unique style (skin color, tops, pants, hairstyle). There are in total 1296 characters, each observed in 72 different poses. For training, we use 1000 characters and we randomly sample 50 poses for each ($N = 50,000$). Auxiliary data for each image frame consists of a character style and a specific pose. The task is to conditionally generate characters not seen during training in different poses.

For the pose part of the auxiliary data, we use a GP-LVM (Lawrence, 2004), similar to what was done in the rotated MNIST experiment for the digit style. Using the GP-LVM also for the character style would not allow us to extrapolate to new character styles during the test phase. To overcome this, we introduce a *representation network*, with which we learn the unobserved parts of the auxiliary data in an amortized way.

Our model easily scales to the size of the SPRITES dataset (time per training epoch: 51.8 ± 0.8 seconds). Moreover, on the test set of 296 characters, our SVGP-VAE achieves a solid performance of 0.0079 ± 0.0009 pixel-wise MSE. In Figure 8, we depict some generations for two test characters. We observe that model faithfully generates the pose information. However, it sometimes wrongly generates parts of the character style. We attribute this to the additional complexity of trying to amortize the learning of the auxiliary data. Extending our initial attempt of using the representation network for such purposes, together with more extensive benchmarking of our model performance, is left for future work. More details on the SPRITES experiment are provided in Appendix A.3.

Figure 8: Conditionally generated SPRITES images for characters not observed during training. Images in the respective upper row are the ground truths, while the images in the respective lower row are conditional generations using our model.

5 Conclusion

We have proposed a novel sparse inference method for GP-VAE models and have shown theoretically and empirically that it is more scalable than existing approaches, while achieving competitive performance. Our approach bridges the gap between sparse variational GP approximations and GP-VAE models, thus enabling the utilization of a large body of work in the sparse GP literature. As such, it represents an important step towards unlocking the possibility to perform amortized GP regression on large datasets with complex likelihoods (e.g., natural images).

Fruitful avenues for future work include considering even more recently proposed sparse GP approaches (Cheng and Boots, 2017; Evans and Nair, 2020) and comparing our proposed scalable GP-VAE solution against other families of deep generative models (Mirza and Osindero, 2014; Eslami et al., 2018a). This would help identify real-world applications where GP-VAEs could be most impactful.

Acknowledgements

M.J. acknowledges funding from the Public Scholarship and Development Fund of the Republic of Slovenia. V.F. was supported by a PhD fellowship from the Swiss Data Science Center and by the grant #2017-110 of the Strategic Focus Area "Personalized Health and Related Technologies (PHRT)" of the ETH Domain. M.J. and V.F. were also supported by ETH core funding (to G.R.). S.M. is supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0021. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research

Projects Agency (DARPA). Furthermore, S.M. was supported by the National Science Foundation under Grants 1928718, 2003237 and 2007719, and by Qualcomm.

References

D. R. Burt, C. E. Rasmussen, and M. Van Der Wilk. Rates of convergence for sparse variational gaussian process regression. *arXiv preprint arXiv:1903.03571*, 2019.

F. P. Casale, A. Dalca, L. Saglietti, J. Listgarten, and N. Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 10369–10380, 2018.

C.-A. Cheng and B. Boots. Variational inference for gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pages 5184–5194, 2017.

C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018.

A. Damianou and N. D. Lawrence. Deep gaussian processes. In *Artificial intelligence and statistics*, pages 207–215. PMLR, 2013.

T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational autoencoders. *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 2:856–865, 2018.

Z. Deng, R. Navarathna, P. Carr, S. Mandt, Y. Yue, I. Matthews, and G. Mori. Factorized variational autoencoders for modeling audience reactions to movies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2577–2586, 2017.

N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360 (6394):1204–1210, 2018a.

S. M. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018b. ISSN 10959203. doi: 10.1126/science.aar6170.

T. W. Evans and P. B. Nair. Quadruply stochastic gaussian processes. *arXiv preprint arXiv:2006.03015*, 2020.

V. Fortuin, G. Dresdner, H. Strathmann, and G. Rätsch. Scalable gaussian processes on discrete domains. *arXiv preprint arXiv:1810.10368*, 2018.

V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt. Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, pages 1651–1661. PMLR, 2020.

D. A. Harville. *Matrix algebra from a statistician's perspective*. Taylor & Francis Group, 1998.

H. V. Henderson and S. R. Searle. On deriving the inverse of a sum of matrices. *Siam Review*, 23(1): 53–60, 1981.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

P. Jähnichen, F. Wenzel, M. Kloft, and S. Mandt. Scalable generalized dynamic topic models. In *International Conference on Artificial Intelligence and Statistics*, pages 1427–1435. PMLR, 2018.

M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2954–2962, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016.

A. Kopf, V. Fortuin, V. R. Somnath, and M. Claassen. Mixture-of-experts variational autoencoder for clustering and generating from similarity-based representations. *arXiv preprint arXiv:1910.07763*, 2019.

N. D. Lawrence. Gaussian Process Latent Variable Models for Visualisation of High Dimensional Data. *Advances in Neural Information Processing Systems*, 2004. doi: 10.1115/OMAE2008-57170.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. Li and S. Mandt. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

W. Lin, N. Hubacher, and M. E. Khan. Variational message passing with structured inference networks. *arXiv preprint arXiv:1803.05589*, 2018.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

M. Pearce. The gaussian process prior vae for interpretable latent dynamics from pixels. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–12, 2020.

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

D. J. Rezende and F. Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

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

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

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.