Gaussian Process Prior Variational Autoencoders

Francesco Paolo Casale[†*], **Adrian V Dalca**[‡§], **Luca Saglietti**[†¶],
Jennifer Listgarten[♯], **Nicolo Fusi**[†]

[†] Microsoft Research New England, Cambridge (MA), USA
[‡] Computer Science and Artificial Intelligence Lab, MIT, Cambridge (MA), USA
[§] Martinos Center for Biomedical Imaging, MGH, HMS, Boston (MA), USA;
[¶] Italian Institute for Genomic Medicine, Torino, Italy
[♯] EECS Department, University of California, Berkeley (CA), USA.
[*] frcasale@microsoft.com

Abstract

Variational autoencoders (VAE) are a powerful and widely-used class of models to learn complex data distributions in an unsupervised fashion. One important limitation of VAEs is the prior assumption that latent sample representations are independent and identically distributed. However, for many important datasets, such as time-series of images, this assumption is too strong and accounting for covariances between samples, such as those in time, can yield to a more appropriate model specification and improve performance in downstream tasks. In this work, we introduce a new model, the Gaussian Process (GP) Prior Variational Autoencoder (GPVAE), to specifically address this issue. The GPVAE aims to combine the power of VAEs with the ability to model correlations afforded by GP priors. To achieve efficient inference in this new class of models, we leverage structure in the covariance matrix, and introduce a new stochastic backpropagation strategy that allows for computing stochastic gradients in a distributed and low-memory fashion. In two image data applications, we show that our method outperforms conditional VAEs (CVAEs) and an adaptation of standard VAEs.

1 Introduction

Dimensionality reduction is a fundamental approach to compression of complex, large-scale data sets, either for visualization or for pre-processing before application of supervised approaches. Auto-encoders represent a rich class of models to perform non-linear dimensionality reduction [45]. Historically, auto-encoders have been framed in one of two modeling camps: the simple and rich capacity language of neural networks; or the probabilistic formalism of generative models, which enables Bayesian capacity control and provides uncertainty over latent encodings. Recently, these two formulations have been combined through the Variational Autoencoder (VAE) [20], wherein the expressiveness of neural networks was used to model both the mean and the variance of a simple likelihood. In these models, latent encodings are assumed to be identically and independently distributed (iid) across both latent dimensions and samples. Despite this simple prior, the model lacks conjugacy, exact inference is intractable and variational inference is used. In fact, the main contribution of the Kingma *et al*., paper is to introduce an improved, general approach for variational inference (also developed in [33]).

One important limitation of the VAE model is the prior assumption that latent representations of samples are iid, whereas in many important problems, accounting for sample structure is crucial for correct model specification and consequently, for optimal results. For example, in autonomous driving, or medical imaging [6, 24], high dimensional images are correlated in time—an iid prior for these would not be sensible because *a priori*, two images that were taken closer in time should have more similar latent representations than images taken further apart. More generally, one can

Preprint. Accepted for poster presentation at NIPS 2018.

have multiple sequences of images from different cars, or medical image sequences from multiple patients. Therefore, the VAE prior should be able to capture multiple levels of correlations at once, including time, object identities, etc. A natural solution to this problem is to replace the VAE iid prior over the latent space with a Gaussian Process (GP) prior [31], which enables the specification of sample correlations through a kernel function [7, 10, 47, 49, 29, 2]. GPs are often amenable to exact inference, and a large body of work in making computationally challenging GP-based models tractable can be leveraged (GPs naively scale cubically in the number of samples) [9, 1, 12, 5, 28, 41].

In this work, we introduce the Gaussian Process Prior Variational Autoencoder (GPVAE), an extension of the VAE latent variable model where sample covariances are modeled through a GP prior on the latent encodings. The introduction of the GP prior, however, introduces two main computational challenges. First, naive learning under the GP prior yields cubic complexity in the number of samples, which is impractical in most applications. To mitigate this problem, one can leverage several tactics commonly used in the GP literature, including the use of pseudo-inputs [5, 9, 12, 28, 41], Kronecker-factorized covariances [8, 29], and low rank structures [3, 22]. Specifically, in the instantiations of GPVAE considered in this paper, we focus on low-rank factorizations of the covariance matrix. A second challenge is that the iid assumption granting the unbiasedness of mini-batch gradient descent (mini-batch GD; commonly used to train standard VAEs) does not hold due to the GP prior, and thus mini-batch GD is no longer applicable. However, for the applications we are interested in, comprising sequences of large-scale images, it is critical from a practical standpoint to avoid processing all samples simultaneously and to somehow regain a procedure that is both low in memory use and yields fast inference. Thus, we propose a new scheme for stochastic full gradient descent that enables full gradient estimation in a distributable and memory-efficient fashion. This is achieved by exploiting the fact that sample covariances are modeled only in the latent (low-dimensional) space, whereas high-dimensional representations are independent when conditioning on the latent ones.

In the next sections we (i) discuss our model in the context of related work, (ii) formally develop the model and the associated inference procedure, (iii) compare GPVAE with alternative models in empirical settings, demonstrating the advantages of our approach.

2 Related work

Our method is related to several extensions of the standard VAE that aim at improving the latent representation by leveraging auxiliary data, such as time annotations, pose information or lighting. An ad hoc attempt to induce structure on the latent space by grouping samples with specific properties in mini-batches was introduced in [21]. More principled approaches proposed a semi-supervised model using a continuous-discrete mixture model that concatenates the input with auxiliary information [18]. Similarly, the conditional VAE [38] incorporates auxiliary information in both the encoder and the decoder, and have been used successfully for sample generation with specific categorical attributes. Building on this approach, several models use the auxiliary information in an undirected way [40, 27, 44, 46, 50].

A separate line of related work aims at designing a more flexible variational posterior distributions, either by considering a dependence on auxiliary variables [25], by allowing structured encoder models [36], or by considering chains of invertible transformations that can produce arbitrarily complex posteriors [19, 26, 32]. In a more principled approach [30, 43], a dependency between latent variables is induced by way of hierarchical structures at the level of the parameters of the variational family.

The extensions of VAEs that are most related to GPVAE, are those that move from the assumption of iid Gaussian prior on the latent representations and consider richer prior distributions [15, 35, 42]. These build on the observation that simplistic priors can induce excessive regularization and become detrimental in unsupervised learning settings [4, 13, 37]. For example, Johnson *et al*., proposed composing latent graphical models with deep observational likelihood. Within their framework, more flexible priors over latent encodings are designed based on conditional independence assumptions, and a conditional random field variational family is used to enable efficient inference via message-passing algorithms [16].

In contrast to existing methods, we propose to model the relationship between the latent space and the auxiliary information using a GP prior, while the encoder and decoder networks project the high-dimensional data to a low-dimensional space and vice versa. Importantly, the proposed approach

allows for modeling arbitrarily complex sample structure in the data. Specifically, in this work, we focus on disentangling sample correlations induced by different aspects of the data. Additionally, GPVAE enables the learning of auxiliary data when these remain unobserved, reusing the approach developed in Gaussian process latent variable models [22]. Finally, using the encoder and decoder networks together with the GP predictive posterior, our model provides a natural framework for out-of-sample predictions of high-dimensional data, enabling data generation for any configuration of the auxiliary data.

3 Gaussian Process Prior Variational Autoencoder

Assume we are given a set of samples (*e. g.,* images), each coupled with different types of auxiliary data (*e. g.,* time, lighting, pose, person identity). In this work, we focus on the case where images of multiple *objects* in different *views* are available. Examples include images of people faces in different poses or images of rotated hand-written digits with different rotation angles. In these problems, we know both which object (person or hand-written digit) is represented in each image in the dataset and in which view (in which pose or rotation angle). Finally, objects and views in the dataset are attached to a feature vector, which we refer to as *object feature vector* and *view feature vector*, respectively. In the face dataset example, object feature vectors will contain face features such as skin color or hair style, while view feature vectors will contain pose features such as polar and azimuthal angles with respect to a reference position. For cases where we do not have such detailed feature vectors, GPVAE allows us to learn them during training.

3.1 Formal description of the model

Let N denote the number of images, P the number of objects and Q the number of possible views. Additionally, let $\{\boldsymbol{y}_n\}_{n=1}^N$ denote K-dimensional samples, $\{\boldsymbol{x}_p\}_{p=1}^P$ M-dimensional object feature vectors for the P objects and $\{\boldsymbol{w}_q\}_{q=1}^Q$ R-dimensional view feature vectors for the Q views. Finally, let $\{\boldsymbol{z}_n\}_{n=1}^N$ denote the L-dimensional latent representations. We consider the following generative process for image data (Fig 1a):

- the latent representation of object p_n in view q_n is generated from object feature vector $\boldsymbol{x}_{p_n}$ and view feature vector $\boldsymbol{w}_{q_n}$ as

$$\boldsymbol{z}_n = f(\boldsymbol{x}_{p_n}, \boldsymbol{w}_{q_n}) + \boldsymbol{\eta}_n, \text{ where } \boldsymbol{\eta}_n \sim \mathcal{N}(\mathbf{0}, \alpha \boldsymbol{I}_L); \tag{1}$$

- image $\boldsymbol{y}_n$ is generated from its latent representation $\boldsymbol{z}_n$ as

$$\boldsymbol{y}_n = g(\boldsymbol{z}_n) + \boldsymbol{\epsilon}_n, \text{ where } \boldsymbol{\epsilon}_n \sim \mathcal{N}\left(\mathbf{0}, \sigma_y^2 \boldsymbol{I}_K\right). \tag{2}$$

The function $f : \mathbb{R}^M \times \mathbb{R}^R \to \mathbb{R}^L$ defines how sample latent representations can be obtained in terms of object and view feature vectors, while $g : \mathbb{R}^L \to \mathbb{R}^K$ projects latent representations to the high-dimensional image space. We consider a Gaussian process (GP) prior on f, which allows us to model sample covariances in the latent space as a function of object and view feature vectors, while we consider a convolutional neural network for g, which is a natural choice for image data [23]. The resulting marginal likelihood of GPVAE, is

$$p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\phi}, \sigma_y^2, \boldsymbol{\theta}, \alpha) = \int p(\boldsymbol{Y} \mid \boldsymbol{Z}, \boldsymbol{\phi}, \sigma_y^2) p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha)\, d\boldsymbol{Z}, \tag{3}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]^T \in \mathbb{R}^{N \times K}$, $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^T \in \mathbb{R}^{N \times L}$, $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_Q]^T \in \mathbb{R}^{Q \times R}$, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P]^T \in \mathbb{R}^{P \times M}$. Additionally, $\boldsymbol{\phi}$ denotes the parameters of g and $\boldsymbol{\theta}$ the GP kernel parameters.

Gaussian Process Model. The GP prior defines the following multivariate normal distribution on latent representations:

$$p(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha) = \prod_{l=1}^{L} \mathcal{N}\left(\boldsymbol{z}^l \mid \mathbf{0}, \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{W}) + \alpha \boldsymbol{I}_N\right), \tag{4}$$

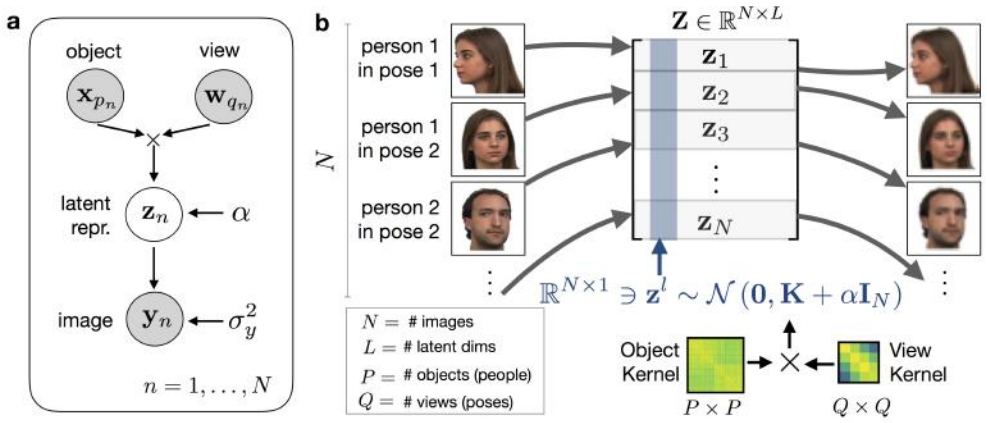

Figure 1: (**a**) Generative model underlying the proposed GPVAE. (**b**) Representation of the inference procedure in GPVAE. Each sample (here an image) is encoded in a low-dimensional space and then decoded to the original space. Covariances between samples modeled through a GP prior on each column of the latent representation matrix $\boldsymbol{Z}$.

where $\boldsymbol{z}^l$ denotes the l-th column of $\boldsymbol{Z}$. In the setting considered in this paper, the covariance function $\boldsymbol{K}_{\boldsymbol{\theta}}$ is composed of a view kernel that models covariances between views and an object kernel that models covariances between objects. Specifically, the covariance between sample n (with corresponding feature vectors $\boldsymbol{x}_{p_n}$ and $\boldsymbol{w}_{q_n}$) and sample m (with corresponding feature vectors $\boldsymbol{x}_{p_m}$ and $\boldsymbol{w}_{q_m}$) is given by the factorized form [2, 29]:

$$\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{W})_{nm} = \mathcal{K}^{(\text{view})}_{\boldsymbol{\theta}}(\boldsymbol{w}_{q_n}, \boldsymbol{w}_{q_m}) \mathcal{K}^{(\text{object})}_{\boldsymbol{\theta}}(\boldsymbol{x}_{p_n}, \boldsymbol{x}_{p_m}). \tag{5}$$

Observed versus unobserved feature vectors Our model can be used when either one of the view or sample feature vectors (or both) remain unobserved. In this setting, we regard the unobserved features as latent variables and obtain a point estimate for them, similar to Gaussian process latent variable models [22].

3.2 Inference

As with a standard VAE, we consider variational inference. Specifically, we consider the following variational distribution over the latent variables

$$q_{\boldsymbol{\psi}}(\boldsymbol{Z} \mid \boldsymbol{Y}) = \prod_n \mathcal{N}\left(\boldsymbol{z}_n \mid \boldsymbol{\mu}^z_{\boldsymbol{\psi}}(\boldsymbol{y}_n), \text{diag}(\boldsymbol{\sigma}^{z\,2}_{\boldsymbol{\psi}}(\boldsymbol{y}_n))\right), \tag{6}$$

which approximates the true posterior on $\boldsymbol{Z}$. In Eq. (6), $\boldsymbol{\mu}^z_{\boldsymbol{\psi}}$ and $\boldsymbol{\sigma}^z_{\boldsymbol{\psi}}$ are the hyperparameters of the variational distribution and are neural network functions of the observed data, while $\boldsymbol{\psi}$ denotes the weights of such neural networks. We obtain the following evidence lower bound (ELBO):

$$\begin{aligned} \log p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\phi}, \sigma_y^2, \boldsymbol{\theta}) \quad \geq \quad & \mathbb{E}_{\boldsymbol{Z} \sim q_{\boldsymbol{\psi}}} \left[\sum_n \log \mathcal{N}(\boldsymbol{y}_n \mid g_{\boldsymbol{\phi}}(\boldsymbol{z}_n), \sigma_y^2 \boldsymbol{I}_K) + \log p\left(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha\right) \right] + \\ & + \frac{1}{2} \sum_{nl} \log(\boldsymbol{\sigma}^{z\,2}_{\boldsymbol{\psi}}(\boldsymbol{y}_n)_l) + \text{const.} \end{aligned} \tag{7}$$

Stochastic backpropagation. We use stochastic backpropagation to maximize the ELBO [20, 33]. Specifically, we approximate the expectation by sampling from a reparameterized variational posterior over the latent representations, obtaining the following loss function:

$$\mathcal{L}\left(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}, \alpha, \sigma_y^2\right) = \underbrace{\frac{1}{K} \sum_n^N \left(\boldsymbol{y}_n - g_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}\,n})\right)^2}_{\text{reconstruction term}} - \frac{\lambda}{L} \left[\underbrace{\log p\left(\boldsymbol{Z}_{\boldsymbol{\psi}} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\theta}, \alpha\right)}_{\text{latent-space GP model}} + \underbrace{\frac{1}{2} \sum_{nl} \log(\boldsymbol{\sigma}^{z\,2}_{\boldsymbol{\psi}}(\boldsymbol{y}_n)_l)}_{\text{regularization}} \right], \tag{8}$$

where λ is a trade-off parameter between the data reconstruction term and the goodness of the latent space GP model, and latent representations $\boldsymbol{Z}_{\boldsymbol{\psi}} = \left[\boldsymbol{z}_{\boldsymbol{\psi}_1}, \ldots, \boldsymbol{z}_{\boldsymbol{\psi}_N}\right] \in \mathbb{R}^{N \times L}$ are sampled as

$$\boldsymbol{z}_{\boldsymbol{\psi}_n} = \mu^z_{\boldsymbol{\psi}}(\boldsymbol{y}_n) + \boldsymbol{\epsilon}_n \odot \sigma^z{}_{\boldsymbol{\psi}}(\boldsymbol{y}_n),\ \boldsymbol{\epsilon}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{L \times L}),\ n = 1, \ldots, N, \tag{9}$$

where $\odot$ denotes the Hadamard product. Full details on the derivation of the loss can be found in Supplementary Information.

Efficient GP parameter inference. Naive parameter inference in Gaussian processes scales cubically with the number of samples [31]. In this work, we achieve linear computations in the number of samples based on the assumption that the overall GP kernel is low-rank, $\boldsymbol{K} = \boldsymbol{V}\boldsymbol{V}^T$ where $\boldsymbol{V} \in \mathbb{R}^{N \times O}$ and $O \ll N$ (Supplementary Information). In order to meet this assumption (i) we exploit that in the applications considered in this work the number of views is low ($Q \ll N$), and (ii) we assume a low-rank form for the object kernel ($M \ll N$, such that $QM \ll N$). Depending on the specific application, different forms of kernels that are amenable to fast inference schemes can be used [29, 39, 48] or alternatively, efficient approximate inference strategies can be employed [12, 28].

Low-memory stochastic backpropagation. The GP prior leads to non-iid observations, making mini-batch gradient descent no longer applicable. On the other hand, however, a naive implementation of full gradient descent is unpractical as it requires loading the entire dataset into memory, which is unfeasible when working with high-dimensional image datasets. To overcome this limitation, we propose the following gradient computation strategy. First, we note that the loss in Eq. 8 is composed by the reconstruction term, which involves high-dimensional observations and factors over individuals, and the GP term, which does not factor over individuals but only involves latent (low-dimensional) representations. This form allows us to employ the following procedure. For the reconstruction term, we can distribute and accumulate the gradients in a mini-batch fashion, thereby removing the need to load up all high-dimensional representations at once. For the GP term, we can compute gradients using all data at once. Since all the computations in this second step take place in the low-dimensional space, the procedure has still low memory requirements. To enable the implementation of this strategy in auto-differentiation-based frameworks, we propose a strategy to compute gradients on the full data based on a local Taylor expansion of the GP term. Full details are given in Supplementary Information.

3.3 Predictive posterior

We derive an approximate predictive posterior for GPVAE that enables out-of-sample predictions of high-dimensional samples. Specifically, given training samples $\boldsymbol{Y}$, object feature vectors $\boldsymbol{X}$, and view feature vectors $\boldsymbol{W}$, the predictive posterior for image representation $\boldsymbol{y}_\star$ of object $p_\star$ in view $q_\star$ is given by

$$p(\boldsymbol{y}_\star \mid \boldsymbol{x}_\star, \boldsymbol{w}_\star, \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{W}) \approx \int \underbrace{p(\boldsymbol{y}_\star \mid \boldsymbol{z}_\star)}_{\text{decode GP prediction}} \underbrace{p(\boldsymbol{z}_\star \mid \boldsymbol{x}_\star, \boldsymbol{w}_\star, \boldsymbol{Z}, \boldsymbol{X}, \boldsymbol{W})}_{\text{latent-space GP predictive posterior}} \underbrace{q(\boldsymbol{Z} \mid \boldsymbol{Y})}_{\text{encode training data}} d\boldsymbol{z}_\star d\boldsymbol{Z} \tag{10}$$

where $\boldsymbol{x}_\star$ and $\boldsymbol{w}_\star$ are object and feature vectors of object $p_\star$ and view $q_\star$ respectively, and we dropped the dependency on parameters for notational compactness. The approximation in Eq. (10) is obtained by replacing the exact posterior on $\boldsymbol{Z}$ with the variational distribution $q(\boldsymbol{Z} \mid \boldsymbol{Y})$ (see Supplementary Information for full details). From Eq. (10), the mean of the GPVAE predictive posterior can be obtained by the following procedure: (i) encode training image data in the latent space through the encoder, (ii) predict latent representation $\boldsymbol{z}_\star$ of image $\boldsymbol{y}_\star$ using the GP predictive posterior, and (iii) decode latent representation $\boldsymbol{z}_\star$ to the high-dimensional image space through the decoder.

4 Experiments

In this work, we focus on the task of making predictions of unseen images. Specifically, we want to predict the image representation of object p in view q in which that object was never observed, assuming that object p was observed in at least one other view. For example, we have previously observed images of the face of a person in several poses and want to predict the image of the face of the same person in a new pose. For a wide range of uses of auto-encoders, this is a good baseline evaluation.

4.1 Compared methods

We compared GPVAE with two alternative optimization strategies and two extensions of the VAE that can be used for the task at hand. Specifically, we considered

- **GPVAE with joint optimization (GPVAE-joint)**, where autoencoder and GP parameters were optimized jointly. We found that convergence was improved by first training encoder/decoder through standard VAE, then optimizing the GP parameters with fixed encoder/decoder for 100 epochs, and finally, optimizing all parameters jointly. Out-of-sample predictions from GPVAE-joint were obtained by using the predictive posterior in Eq. (10);
- **GPVAE with disjoint optimization (GPVAE-dis)**, where we first learned the encoder/decoder parameters through standard VAE, and then optimized the GP parameters with fixed encoder/decoder. Again, out-of-sample predictions were obtained by using the predictive posterior in Eq. (10);
- **Conditional VAE (CVAE)** [38], where view auxiliary information was provided as input to both the encoder and decoder networks (Figure S1, S2). After training, we considered the following procedure to generate an image of object p in view q. First, we computed latent representations of all the images of object p across all the views in the training data (in this setting, CVAE latent representations are supposedly independent from the view). Second, we averaged all the obtained latent representations to obtain a unique representation of object p. Finally, we fed the latent representation of object p together with out-of-sample view q to the CVAE decoder. As an alternative implementation, we also tried to consider the latent representation of a random image of object p instead of averaging, but the performance was worse;
- **Linear Interpolation in VAE latent space (LIVAE)**, which uses linear interpolation between observed views of an object in the latent space learned through standard VAE in order to predict unobserved views of the same object. Specifically, denoting $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ as the latent representations of images of a given object in views r_1 and r_2, a prediction for the image of that same object in an intermediate view, $r_\star$, is obtained by first linearly interpolating between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, and then projecting the interpolated latent representation to the high-dimensional image space.

Consistently with the L2 reconstruction error in the GPVAE loss in Eq. (8), we considered pixel-wise mean squared error (MSE) as evaluation metric for all compared methods. We used the same architecture for encoder and decoder neural networks in each variation of the VAE (see Figure S1, S2 in Supplementary Information). The architecture and the hyper-parameter λ, were chosen to minimize the ELBO loss for the standard VAE on a validation set (Figure S3). For CVAE and LIVAE, we also considered the alternative strategy of selecting the value of the trade-off parameter that maximizes out-of-sample prediction performance on the validation set (the results for these two methods are in Figure S4 and S5). All models were trained using the Adam optimizer [17] with standard parameters and a learning rate of 0.001. When optimizing GP parameters with fixed encoder/decoder, we observed that higher learning rates led to faster convergence without any loss in performance, and thus we used a higher learning rate of 0.01 in this setting.

4.2 Rotated MNIST

Setup. We considered a variation of the MNIST dataset, consisting of rotated images of handwritten "3" digits with different rotation angles. In this setup, objects correspond to different draws of the digit "3" while views correspond to different rotational states. View features are observed scalars, corresponding to the attached rotation angles. Conversely, object feature vectors are unobserved and learned from data—no draw-specific features are available.

Dataset generation. We generated a dataset from 400 handwritten versions of the digit three by rotating through $Q = 16$ evenly separated rotation angles in $[0, 2\pi)$, for a total of $N = 6,400$ samples. We then kept 90% of the data for training and test, and the rest for validation. From the training and test sets, we then randomly removed 25% of the images to consider the scenario of incomplete data. Finally, the set that we used for out-of-sample predictions (test set) was created by removing one of the views (*i. e.,* rotation angles) from the remaining images. This procedure resulted in $4,050$ training images spanning 15 rotations and 270 test images spanning one rotation.

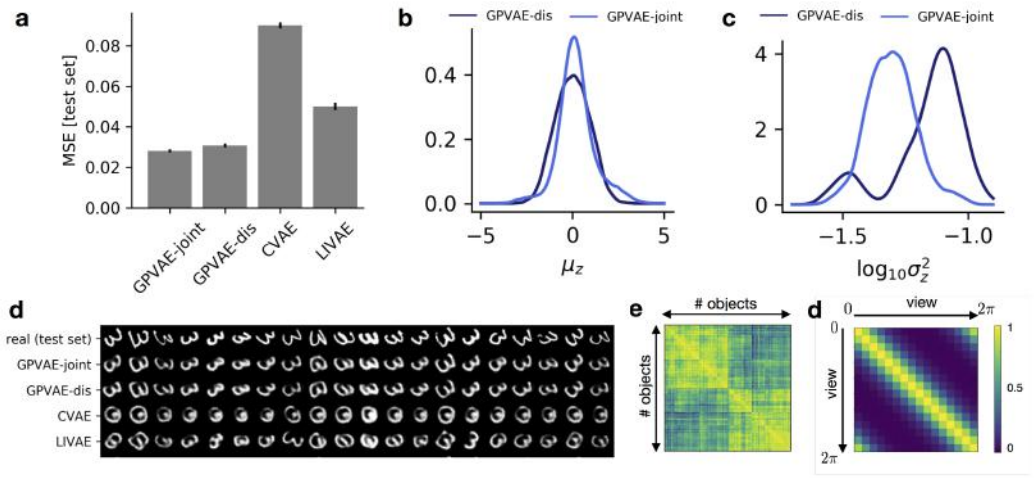

Figure 2: Results from experiments on rotated MNIST. (**a**) Mean squared error on test set. Error bars represent standard error of per-sample MSE. (**b**) Empirical density of estimated means of q_ψ, aggregated over all latent dimensions. (**c**) Empirical density of estimated log variances of q_ψ. (**d**) Out-of-sample predictions for ten random draws of digit "3" at the out-of-sample rotation state. (**e**, **f**) Object and view covariances learned through GPVAE-joint.

Autoencoder and GP model. We set the dimension of the latent space to $L = 16$. For encoder/decoder neural networks we considered the convolutional architecture in Figure S1. As view kernel, we considered a periodic squared exponential kernel taking rotation angles as inputs. As object kernel, we considered a linear kernel taking the object feature vectors as inputs. As object feature vectors are unobserved, we learned from data—their dimensionality was set to $M = 8$. The resulting composite kernel $\boldsymbol{K}$, expresses the covariance between images n and m in terms of the corresponding rotations angles w_{q_n} and w_{q_m} and object feature vectors $\boldsymbol{x}_{p_n}$ and $\boldsymbol{x}_{p_m}$ as

$$\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{w})_{nm} = \beta \underbrace{\exp\left(-\frac{2\sin^2|w_{q_n} - w_{q_m}|}{\nu^2}\right)}_{\text{rotation kernel}} \cdot \underbrace{\boldsymbol{x}_{p_n}^T \boldsymbol{x}_{p_m}}_{\text{digit draw kernel}}, \tag{11}$$

where $\beta \geq 0$ and $\nu \geq 0$ are kernel hyper-parameters learned during training of the model [31], and we set $\boldsymbol{\theta} = \{\beta, \nu\}$.

Results. GPVAE-joint and GPVAE-dis yielded lower MSE than CVAE and LIVAE in the interpolation task, with GPVAE-jointperforming significantly better than GPVAE-dis(0.0280 ± 0.0008 for GPVAE-joint vs 0.0306 ± 0.0009 for GPVAE-dis, $p < 0.02$, **Fig. 2a**,**b**). Importantly, GPVAE-joint learns different variational parameters than a standard VAE (Fig. 2c,d), used also by GPVAE-dis, consistent with the fact that GPVAE-joint performs better by adapting the VAE latent space using guidance from the prior.

4.3 Face dataset

Setup. As second application, we considered the Face-Place Database (3.0) [34], which contains images of people faces in different poses. In this setting, objects correspond to the person identities while views correspond to different poses. Both view and object feature vectors are unobserved and learned from data. The task is to predict images of people face in orientations that remained unobserved.

Data. We considered 4,835 images from the Face-Place Database (3.0) [34], which includes images of faces for 542 people shown across nine different poses (frontal and 90, 60, 45, 30 degrees left

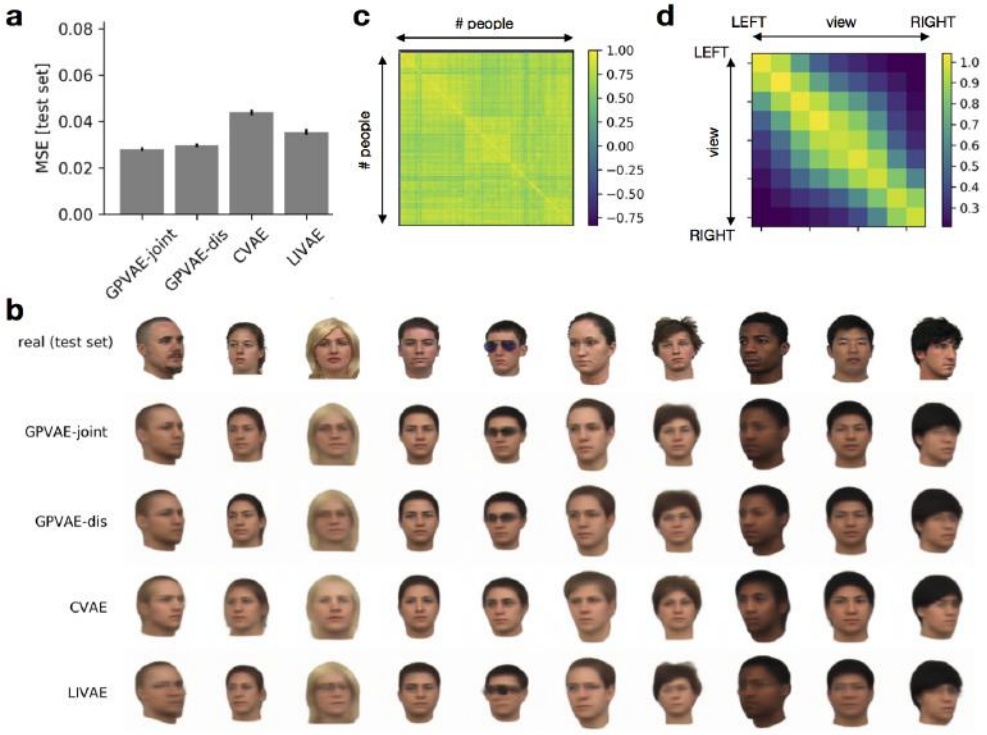

Figure 3: Results from experiments on the face dataset. (**a**) Mean squared error on test set (**b**) Out-of-sample predictions for ten random people faces in out-of-sample poses. (**c**, **d**) Object and view covariances learned through GPVAE-joint.

and right[1]). We randomly selected 80% of the data for training ($n = 3,868$), 10% for validation ($n = 484$) and 10% for testing ($n = 483$). All images were rescaled to 128×128.

Autoencoder and GP model. We set the dimension of the latent space to $L = 256$. For encoder/decoder neural networks we considered the convolutional architecture in Figure S2. We consider a full-rank covariance as a view covariance (only nine poses/views are present in the dataset) and a linear covariance for the object covariance ($M = 64$).

Results. GPVAE-joint and GPVAE-dis yielded lower MSE than CVAE and LIVAE (**Fig. 2a**,**b**). In contrast to the MNIST problem, the difference between GPVAE-joint andGPVAE-dis was not significant (0.0281 ± 0.0008 for GPVAE-joint vs 0.0298 ± 0.0008 for GPVAE-dis). Importantly, GPVAE-joint was able to dissect people (object) and pose (view) covariances by learning people and pose kernel jointly (**Fig. 2a**,**b**).

5 Discussion

We introduced GPVAE, a generative model that incorporates a GP prior over the latent space. We also presented a low-memory and computationally efficient inference strategy for this model, which makes the model applicable to large high-dimensional datasets. GPVAE outperforms natural baselines (CVAE and linear interpolations in the VAE latent space) when predicting out-of-sample test images of objects in specified views (*e. g.,* pose of a face, rotation of a digit). Possible future work includes augmenting the GPVAE loss with a discriminator function, similar in spirit to a GAN[11], or changing the loss to be perception-aware [14] (see results from preliminary experiments in Figure S6).

[1] We could have used the pose angles as view feature scalar similar to the application in rotated MNIST, but purposely ignored these features to consider a more challenging setting were neither object and view features are observed.

Acknowledgments

Stimulus images courtesy of Michael J. Tarr, Center for the Neural Basis of Cognition and Department of Psychology, Carnegie Mellon University, http:// www.tarrlab.org. Funding provided by NSF award 0339122.

References

[1] M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding probabilistic sparse gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–1541, 2016.

[2] E. V. Bonilla, F. V. Agakov, and C. K. Williams. Kernel multi-task learning using task-specific features. In *Artificial Intelligence and Statistics*, pages 43–50, 2007.

[3] F. P. Casale, B. Rakitsch, C. Lippert, and O. Stegle. Efficient set tests for the genetic analysis of correlated traits. *Nature methods*, 12(8):755, 2015.

[4] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[5] L. Csató and M. Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

[6] A. V. Dalca, R. Sridharan, M. R. Sabuncu, and P. Golland. Predictive modeling of anatomy with genetic and clinical data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 519–526. Springer, 2015.

[7] N. Durrande, D. Ginsbourger, O. Roustant, and L. Carraro. Additive covariance kernels for high-dimensional gaussian process modeling. *arXiv preprint arXiv:1111.6233*, 2011.

[8] N. Fusi and J. Listgarten. Flexible modelling of genetic effects on function-valued traits. In M. Singh, editor, *Research in Computational Molecular Biology: 20th Annual Conference, RECOMB 2016, Santa Monica, CA, USA, April 17-21, 2016, Proceedings*, pages 95–110. Springer International Publishing, 2016.

[9] 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.

[10] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of machine learning research*, 12(Jul):2211–2268, 2011.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, page 282. Citeseer, 2013.

[13] M. D. Hoffman and M. J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

[14] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1133–1141. IEEE, 2017.

[15] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[16] M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

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

[18] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[19] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

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

[21] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[22] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(Nov):1783–1816, 2005.

[23] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[24] J. Lonsdale, J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580, 2013.

[25] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[26] E. Nalisnick, L. Hertel, and P. Smyth. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2, 2016.

[27] G. Pandey and A. Dukkipati. Variational methods for conditional multimodal deep learning. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 308–315. IEEE, 2017.

[28] 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.

[29] B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. In *Advances in neural information processing systems*, pages 1466–1474, 2013.

[30] R. Ranganath, D. Tran, and D. Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[31] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

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

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

[34] G. Righi, J. J. Peissig, and M. J. Tarr. Recognizing disguised faces. *Visual Cognition*, 20(2):143–169, 2012.

[35] R. Shu, J. Brofos, F. Zhang, H. H. Bui, M. Ghavamzadeh, and M. Kochenderfer. Stochastic video prediction with conditional density estimation. In *ECCV Workshop on Action and Anticipation for Visual Learning*, volume 2, 2016.

[36] N. Siddharth, B. Paige, A. Desmaison, V. de Meent, F. Wood, N. D. Goodman, P. Kohli, P. H. Torr, et al. Inducing interpretable representations with variational autoencoders. *arXiv preprint arXiv:1611.07492*, 2016.

[37] N. Siddharth, B. Paige, J.-W. Van de Meent, A. Desmaison, F. Wood, N. D. Goodman, P. Kohli, and P. H. Torr. Learning disentangled representations with semi-supervised deep generative models. *ArXiv e-prints (Jun 2017)*, 2017.

[38] 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.

[39] O. Stegle, C. Lippert, J. M. Mooij, N. D. Lawrence, and K. M. Borgwardt. Efficient inference in matrix-variate gaussian models with\iid observation noise. In *Advances in neural information processing systems*, pages 630–638, 2011.

[40] M. Suzuki, K. Nakayama, and Y. Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

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

[42] J. M. Tomczak and M. Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

[43] D. Tran, R. Ranganath, and D. M. Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

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

[45] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[46] W. Wang, X. Yan, H. Lee, and K. Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016.

[47] A. Wilson and R. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, 2013.

[48] A. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

[49] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

[50] M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning. *arXiv preprint arXiv:1802.05335*, 2018.