Gaussian Process Encoders: VAEs with Reliable Latent-Space Uncertainty

Judith Bütepage[0000−0001−5344−8042] ✉, Lucas Maystre[0000−0002−8307−7673], and Mounia Lalmas[0000−0002−3531−3096]

[1] Spotify
[2] {judithb, lucasm, mounial}@spotify.com

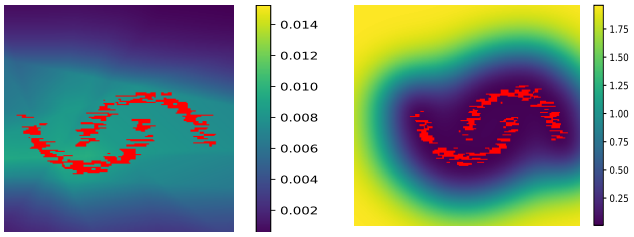
Abstract. Variational autoencoders are a versatile class of deep latent variable models. They learn expressive latent representations of high dimensional data. However, the latent variance is not a reliable estimate of how uncertain the model is about a given input point. We address this issue by introducing a sparse Gaussian process encoder. The Gaussian process leads to more reliable uncertainty estimates in the latent space. We investigate the implications of replacing the neural network encoder with a Gaussian process in light of recent research. We then demonstrate how the Gaussian Process encoder generates reliable uncertainty estimates while maintaining good likelihood estimates on a range of anomaly detection problems. Finally, we investigate the sensitivity to noise in the training data and show how an appropriate choice of Gaussian process kernel can lead to automatic relevance determination.

Keywords: Variational Autoencoder · Uncertainty estimation · Anomaly detection · Gaussian Process

1 Introduction

Generative models can represent a joint probability distribution over observed and latent variables. Modern generative models often combine the representational power of deep neural networks with the structured representations encoded by probabilistic graphical models [12]. One popular class of deep latent variable models are Variational Autoencoders (VAEs) [14, 22]. VAEs generate samples of the data distribution by transforming a sample from a simple noise distribution, the prior, into an output distribution in data space with the help of a neural network (NN), the decoder network. To determine the latent variable distribution for a given data point, an encoder network, representing the approximate posterior, is used to determine the form of the latent variable of each data point. VAEs are trained using the Evidence Lower BOund (ELBO), which regularizes the data likelihood under the approximate posterior with the Kullback-Leibler divergence (KL) between the approximate posterior and a prior distribution.

While this inference scheme usually works well for the mean parameter of the latent variable, it often fails to learn an informative variance parameter for each data point [3]. In many cases, the latent variance fails to correlate with



(a) VAE (b) GP-VAE

Fig. 1. The latent variance of a (a) VAE encoder and (b) GP-VAE encoder trained on the two-moon dataset and evaluated over a grid of points around the training data points. The red dots visualize the latent mean of the training data.

how uncertain the model is about the input. An example of this is shown in Figure 1(a), which depicts the latent variance estimates of a VAE trained on the two-moon dataset [15]. Contrary to expectation, the uncertainty is decaying the further away from the training data we evaluate.

This behavior becomes problematic when one relies on the estimates of the latent uncertainty. For example in reinforcement learning, when sampling in the latent space of a temporal VAE to predict the next observation given an uncertain input, it is important to sample a variety of possible futures [7]. Note that the problem of modelling latent uncertainty is different to modelling an accurate variance in the data space using the decoder as discussed in e.g. [24]. The output uncertainty of a VAE centers around a single data point, while a high latent variance produces a larger variety of samples. Another example that requires reliable estimates of the latent uncertainty is anomaly or out-of-distribution (OOD) detection using generative models. As demonstrated in [9, 17] the likelihood distribution of a VAE cannot reliably detect OOD data. Variance estimates in the latent space can be an alternative.

In this work, we extend VAEs to enable reliable latent uncertainty estimates for in-distribution (ID) and OOD data. We replace the neural network encoder traditionally used in VAEs with a Gaussian process (GP) encoder (Section 2.1). We refer to this model as a GP-VAE. Our formulation considers the GP encoder as a drop-in replacement of neural networks. With this, we retain the versatility of VAEs while gaining several advantages: reliable uncertainty estimates, reduced overfitting, and increased robustness to noise. For scalability, we parameterize the GP by using a small number of inducing points, similar to sparse variational GPs [26]. This enables us to learn a compact mapping from the data space to the latent space.

The GP encoder learns to represent data points in the latent space based on their similarity, by using a kernel function. It produces principled uncertainty estimates as shown for the moon dataset in Figure 1(b). In contrast to the standard VAE in Figure 1(a), the latent variance of the GP-VAE increases with the distance to the latent means of the training data.

We evaluate our GP-VAE model both in terms of how well it fits the ID data distribution as well as how informative the latent variance is compared to baselines (Sections 4.1 – 4.2). To test the reliability of the latent variance, we evaluate our proposed model on a variety of anomaly detection tasks (Section 4.3). We also show that the model's ability to identify OOD data is robust to OOD noise in the training data (Sections 4.4). We further demonstrate how the reliable latent variance estimates of the GP-VAE allow for meaningful synthesized variants of encoded data (Section 4.5). Finally, we demonstrate how a specific choice of the GP kernel, namely an additive kernel, leads to interpretable models and allows us to identify which input features are important to distinguish between ID and OOD data (Section 4.6).

1.1 Contributions

Our contributions are threefold: *a*) we introduce a Gaussian process encoder for VAEs that infers reliable uncertainty estimates in the latent space of a VAE; *b*) we derive a scalable inference scheme for the GP encoder using a set of inducing points; and finally *c*) we describe how to use additive kernels to create interpretable models that can be used to identify features that distinguish ID from OOD data.

2 Background

We begin by introducing the general ideas behind variational autoencoders. We then discuss why the common choice of a neural network encoder leads to poor latent uncertainty estimates. In Section 3.2 we relate back to this section and discuss the implications of replacing the NN with a GP encoder after having introduced the GP encoder formally in Section 3.1.

2.1 Variational Autoencoder

Let $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of N data points, where $\boldsymbol{x}_n \in \mathbf{R}^K$, and $\boldsymbol{X} \in \mathbf{R}^{N \times K}$ be the collection of data points as a stacked matrix. We construct a generative model of the data with parameters $\boldsymbol{\theta}$ that maximizes the data log-likelihood $\log p_{\boldsymbol{\theta}}(\boldsymbol{X})$. We follow the same generative model as assumed for VAEs, i.e., that each data point $\boldsymbol{x}$ is generated independently and is conditioned on a latent variable $\boldsymbol{z} \in \mathbf{R}^D$, where typically $D \ll K$.

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \prod_n \int p_{\boldsymbol{\theta}}(\boldsymbol{x}_n \mid \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}. \tag{1}$$

As commonly assumed for VAEs, we assume an i.i.d. zero-mean and spherical Gaussian prior for the latent variables $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \sigma^2 \boldsymbol{I})$, and model the conditional likelihood of the data by using $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}[\mu_{\boldsymbol{\theta}}(\boldsymbol{z}), \sigma^2_{\boldsymbol{\theta}}(\boldsymbol{z})]$, where $\mu_{\boldsymbol{\theta}}(\cdot)$ and $\sigma^2_{\boldsymbol{\theta}}(\cdot)$ are feed-forward neural networks. The choice of the likelihood

distribution depends on the dataset and does not need to be Gaussian. When generating the data, we first sample a latent variable from the prior $p(\boldsymbol{z})$ and subsequently sample a data point using the decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})$.

Directly maximizing Equation 1 (or the log thereof) is intractable. Therefore, following the derivation of VAEs, we use Jensen's inequality to derive the evidence lower-bound

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{X}) \geq \sum_n \left\{ \mathbf{E}_{q(\boldsymbol{z}|\boldsymbol{x}_n)} \left[\log p_\theta(\boldsymbol{x}_n \mid \boldsymbol{z}) \right] - \mathrm{KL}[q(\boldsymbol{z} \mid \boldsymbol{x}_n) \| p(\boldsymbol{z})] \right\}, \tag{2}$$

where $q(\boldsymbol{z} \mid \boldsymbol{x})$ is an auxiliary distribution that approximates the posterior distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$ [14]. This variational distribution is commonly chosen to be a Gaussian with diagonal covariance where the mean and covariance matrix are functions of the input data point:

$$q(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}\left\{ \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathrm{diag}[\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x})] \right\}. \tag{3}$$

These functions are paramterized by $\boldsymbol{\phi}$. As such, we can think of the functions $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\cdot)$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\cdot)$ as *encoding* the data point into the latent space. In the literature on VAEs, these functions are usually chosen to be neural networks, as illustrated in Figure 3a).

Training a VAE entails maximizing Equation (2) over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. In contrast to Equation (1), this can be done efficiently, provided that the encoder and decoder are differentiable. We refer to [13] for more background on VAEs.

2.2 Latent Variance Estimates of NN

Neural Network encoders can exhibit different learning behaviors when optimizing the ELBO. For example, one common phenomenon occurs when the KL divergence in the ELBO is too strong in the early stages of training, which then forces the approximate posterior to be equal to the prior. This can impact either all latent dimensions, called posterior collapse [8], or a subset of latent dimensions, called the dying units problem [31].

A second behavior discussed by [3] is that the ELBO pushes both the encoder and decoder variance to zero to achieve minimal reconstruction errors. Consider the KL divergence part of the ELBO in Equation (2). In the case of Gaussian latent variables with a standard normal prior we have $\mathrm{KL}[q(\boldsymbol{z} \mid \boldsymbol{x}_n) \| p(\boldsymbol{z})] = \frac{1}{2} \sum_{d=1}^{D} [\boldsymbol{\sigma}^2_{d,\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{\mu}_{d,\boldsymbol{\phi}}(\boldsymbol{x})^2 - 1 - \log(\boldsymbol{\sigma}_{d,\boldsymbol{\phi}}(\boldsymbol{x})^2)]$. The part of the KL concerned with the latent variance rarely dominates the ELBO even when the latent variance values are relatively small. Therefore, a NN encoder can neglect that part of the KL divergence and set the latent variance to very small values. This allows the model to focus on maximizing the expected log likelihood while simultaneously minimizing the distance of the latent means and the prior mean as dictated by the KL, i.e. $\boldsymbol{\mu}_{d,\boldsymbol{\phi}}(\boldsymbol{x})^2$. We can view this vanishing latent variance as a form of overfitting. This behavior is illustrated in Figure 2 a), which depicts 250 encoded mean and variance values of a VAE trained on the FashionMnist dataset [30]. The latent mean is clustered around the prior mean, namely zero, while the latent variance is too small to be visible.

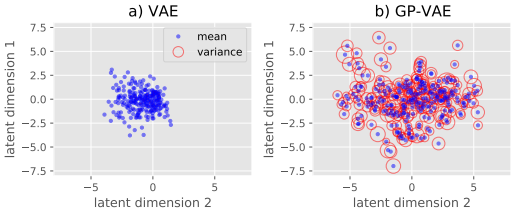


Fig. 2. Latent space mean (blue) and variance (red) of 250 data points of the FashionMnist dataset. **a)** The latent variance of the VAE is too small to be visible. **b)** The GP-VAE spreads its probability mass to accommodate larger latent variance estimates.

2.3 Mismatch between the Prior and Approximate Posterior

The preference of the VAE to minimize the reconstruction error can lead to inaccurate inference, where the approximate posterior aggregated over all training data points is no longer equal to the prior. When this is the case, sampling from the prior will not reconstruct the entire data distribution. The authors of [3] propose a two-stage approach that consists of a standard VAE and a second generative model trained to emulate samples from the aggregated approximate posterior. Instead of a two-stage solution, it has also been proposed to approximate the aggregated posterior by a mixture of variational posteriors with pseudo-inputs during training [27].

An alternative solution to the misalignment between prior and the aggregated approximate posterior is proposed by [32] who suggest to add an additional loss term, namely the mutual information between the data and the latent variables. In their experiments, they remove the KL term from the ELBO altogether and implement the mutual information as the Maximum Mean Discrepancy (MMD). They show that the aggregated approximate posterior of their model, the InfoVAE, is closer to the prior compared to the VAE's. However, this is only attributable to the latent mean, not the latent variance which converges to zero. Since the MMD is only concerned with samples from the latent distribution and not the latent variance parameters, the InfoVAE is free to set the latent variance essentially to zero.

3 Methodology

We suggest to use the predictive equations of a Gaussian process to parametrize the encoder. At a high level, we define the the encoder as the posterior distribution of a Gaussian process, given a small number of pseudo-observations that we treat as parameters. The variance in the latent space is thus estimated in a principled way, by using Bayes' rule in a well-defined probabilistic model. In contrast to using a neural network encoder, using a GP encoder leads to inductive biases that prevent the variance from vanishing.

In this section we describe how we replace a neural network encoder with a sparse Gaussian process. In short, we learn a number of representative data

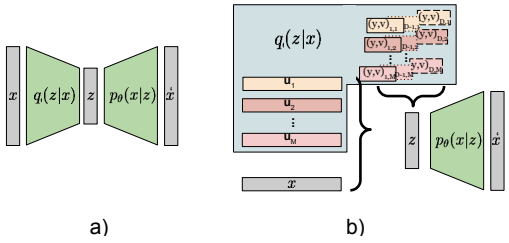


Fig. 3. Model structure: **a)** The usual structure of a VAE with neural networks as encoder and decoder. **b)** Our proposed VAE with a sparse GP as encoder and a neural network as decoder.

points, so called inducing points, and their corresponding value in the latent space. This allows us to employ a Gaussian process to infer the latent distribution for ID data points, which in turn can be decoded by a neural network into the data space.

3.1 Gaussian Process Encoder

In contrast to the common formulation of VAEs, we propose using a sparse Gaussian process encoder instead of a neural network, as depicted in Figure 3b). We can view the encoder as the output of an auxiliary Gaussian process recognition model $\hat{\boldsymbol{z}}(\boldsymbol{x})$ that maps from the data space to the latent space. We assume, for simplicity, that the D dimensions of the multivariate GP's output are a priori identically and independently distributed, i.e.,

$$\hat{\boldsymbol{z}}(\boldsymbol{x}) \sim \mathcal{GP}[\boldsymbol{0}, k(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{I}], \tag{4}$$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ is a positive-definite kernel. We assume that the correlation between data points can be captured by a set of M pseudo-inputs or inducing points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M \in \mathbf{R}^K$. Suppose that, for each inducing point $m = 1, \ldots, M$, we observe that, at input $\boldsymbol{u}_m$, the GP has value $\boldsymbol{y}_m \in \mathbf{R}^D$ with Gaussian measurement noise $\boldsymbol{v}_m \in \mathbf{R}^D_{>0}$. Then, we can formalize the distribution of $\hat{\boldsymbol{z}}$ at a new input $\boldsymbol{x}$ as the predictive distribution of a GP.

To this end, we stack these vectors into matrices $\boldsymbol{Y}, \boldsymbol{V} \in \mathbf{R}^{M \times D}$ and denote by $\boldsymbol{y}^d$ and $\boldsymbol{v}^d$ the dth column of the respective matrix. Finally, let $\boldsymbol{K} = [k(\boldsymbol{u}_m, \boldsymbol{u}_{m'})]$ be the $M \times M$ matrix obtained by evaluating the kernel at each pair of inducing points.

Given these, we define the dth dimension of the encoder (Equation (3)) by

$$\begin{aligned} \mu_{d,\boldsymbol{\phi}}(\boldsymbol{x}) &= \boldsymbol{k}^\top \left[\boldsymbol{K} + \operatorname{diag}(\boldsymbol{v}^d)\right]^{-1} \boldsymbol{y}^d, \\ \sigma^2_{d,\boldsymbol{\phi}}(\boldsymbol{x}) &= k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^\top \left[\boldsymbol{K} + \operatorname{diag}(\boldsymbol{v}^d)\right]^{-1} \boldsymbol{k}, \end{aligned} \tag{5}$$

where $\boldsymbol{k} = [k(\boldsymbol{x}, \boldsymbol{U})]$. As in Equation (3), we therefore have can formalize the encoder as $q(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \operatorname{diag}[\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x})])$, where $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x})$ are the stacked outputs of Equation (5). The parameters of the encoder are given by

$\boldsymbol{\phi} = \{\boldsymbol{U}, \boldsymbol{Y}, \boldsymbol{V}\}$, to which we can add hyperparameters of the kernel function. Since the functions in Equation (5) are differentiable with respect to $\boldsymbol{\phi}$, we can use them to replace the neural networks typically used in the literature. The VAE is trained as previously, by optimizing the evidence lower bound over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

Optimizing the parameters of the predictive distribution of a sparse GP directly as we do here has recently been proposed by [10]. The authors show that Parametric Predictive GP Regression exhibits significantly better calibrated uncertainties than e.g. Fully Independent Training Conditional approximations [10, 25]

Aggregated approximated posterior. As discussed in Section 2.3, to sample from VAEs effectively, we need to make use of the aggregated approximated posterior, when it is not equal to the prior. Our formulation of a sparse VAE lends itself to represent the aggregated approximated posterior using the inducing points as

$$\hat{q}(\boldsymbol{z}) = \frac{1}{M} \sum_m q(\boldsymbol{z} \mid \boldsymbol{u}_m) = \frac{1}{M} \sum_m \mathcal{N}(\boldsymbol{y}_m, \mathrm{diag}(\boldsymbol{v}_m)). \tag{6}$$

This formulation is similar to the aggregated approximated posterior using pseudo-inputs proposed by [27]. Contrary to a neural network encoder, the sparse GP learns these pseudo-inputs in form of the inducing points automatically.

3.2 The Implications of a Gaussian Process Encoder

As discussed in Sections 2.2, neural networks can fail to learn reliable uncertainty estimates. The latent variance of a GP encoder on the other hand reflects how close a data point is to the training data or inducing points. It is therefore constraint in its ability to set the latent variance to zero. To accommodate reliable latent variance estimates while minimizing the reconstruction error, the GP-VAE needs to minimize both the latent mean and the latent variance part of the KL. The loss connected to the variance will be smaller, allowing the KL loss connected to the mean to be larger, i.e. the latent mean spreads more than in the case of a VAE. As shown in Figure 2 b), the GP-VAE trained on the FashionMnist dataset spreads the probability mass of the approximate posterior more than encouraged by the prior, while maintaining reliable latent variance estimates. Thus, to reconstruct the data distribution we cannot sample from the prior, falling back to the problem discussed in Section 2.3. In contrast to a two-stage solution however, the use of inducing points in the GP encoder allows us to formulate an aggregated approximate posterior without the need of fitting a separate model (see Section 3.1).

3.3 Out-Of-Distribution Detection

Given a test dataset $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{N_t}^*\}$, we want to evaluate whether the latent variance can be used to determine whether a data point has been drawn from

the same distribution as the training data or is a OOD data point. To this end, we average over the D values of the diagonal of covariance matrix $\boldsymbol{\sigma}_n^* \doteq \frac{1}{D}\sum_{d=1}^{D} \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\boldsymbol{x}_n^*)^d$ of the Gaussian approximate posterior $q(\boldsymbol{z} \mid \boldsymbol{x}_n^*)$ in Equation (5). These values can be interpreted as a measure of how uncertain the model is about the data point and should therefore distinguish between ID and OOD test data. In the experiments, we report the area under the Receiver Operating Characteristics *roc* and Precision-Recall curve *pr* for $\{\boldsymbol{\sigma}_1^*, \ldots, \boldsymbol{\sigma}_{N_t}^*\}$. To use the values for OOD detection in practice, one needs to determine an appropriate threshold, e.g. with a small number of labeled data points.

4 Experiments

We evaluate the GP-VAE both in terms of how well it models the training data distribution as well as how reliable the estimates of the latent variance are compared to other models. We evaluate our approach on a number of OOD detection datasets (Sections 4.1–4.3). To test for the robustness of the latent variance, we introduce OOD examples into the training ID data and analyze how this noise impacts OOD detection on the test data (Section 4.4). We also demonstrate that informative uncertainty estimates can be used to generate a diverse set of data samples (Section 4.5). Finally, we describe how the choice of an additive kernel function leads to an interpretable model that allows identifying the input features contributing to OOD detection (Section 4.6).

Datasets To compare the ability of our model to detect OOD data samples, we run extensive experiments on the Outlier Detection DataSets (ODDS) Library [20]. This library contains a number of datasets from different domains. We test our approach on 20 datasets in the ODDS library. Each dataset consists of ID data points and OOD data points. We create an ID training set by randomly selecting half of the ID data points and test on the remaining ID data points and the OOD data points. The exact specifics for the datasets are described in the Appendix. We also test our approach on an image dataset, namely the FashionMnist (FM) [30] and Mnist (M) [16] dataset, as done in [17].

Model specifics We compare our GP-VAE model to baselines trained under the same conditions as ours. The two generative models that we compare to are the standard VAE and the InfoVAE [32]. The encoder and decoder neural networks of all VAE-based models consist of two fully connected layers and one latent variable layer. As a kernel function for the GP-VAE, we use the squared exponential kernel. More specifics on the model architecture and training protocols can be found in the Appendix. Additionally, we train a recent supervised approach based on deviation networks (Dev-Net) [19] on all datasets as a state-of-the-art baseline for OOD detection. We evaluate the models after training the VAE-based models for 200 epochs and the Dev-Net for 50 epochs and repeat the experiment with five different seeds.

4.1 Log Likelihood

We start by comparing the ability to model the ID data distribution by computing the log likelihood for all test datasets. We approximate the log likelihood with the help of importance sampling by

$$\log(p(\boldsymbol{x}_n)) = \log\left(\mathbf{E}_{q(\boldsymbol{z}|\boldsymbol{x}_n)}\left[p_\theta(\boldsymbol{x}_n \mid \boldsymbol{z})\frac{p(\boldsymbol{z})}{q(\boldsymbol{z} \mid \boldsymbol{x}_n)}\right]\right). \tag{7}$$

As shown in the first three columns in Table 1, the GP-VAE performs similar to the VAE and InfoVAE in terms of likelihood estimates. The small latent variance values of the VAE and InfoVAE result in high denominator values in the above equation, which can impact their likelihood estimate negatively. The standard deviations across runs and reconstruction errors of all three models for the OOD datasets are presented in the Appendix.

4.2 Uncertainty in the Latent Space

We introduced the GP encoder as a means to reliably express uncertainty about unfamiliar data points. A model should be more uncertain about data points that have low similarity to the training data compared to more typical examples. Thus, we expect a positive correlation between the average distance of a data point to all training data points and its latent variance. To test how well the different VAE-based models follow this behavior, we train an VAE, InfoVAE and GP-VAE on all 20 datasets selected from the ODDS. We then compute the average euclidean distance of each ID and OOD test data point to all data points in the ID training dataset and infer the latent variance values using the specific encoders. Finally we compute the Pearson correlation coefficient (PCC) between the distances and the latent variances. As shown in Figure 4, both the VAE and InfoVAE fail to capture the similarity of a data point to the training data as the average correlation is PCC=-0.069 and PCC=-0.0019 respectively. In contrast, the GP-VAE correlates positively with an average of PCC=0.48.

4.3 Benchmarking OOD Detection

To test whether the latent variance reliably indicates if a data point is ID or OOD, we use the latent variance values for OOD detection. We repeat model training

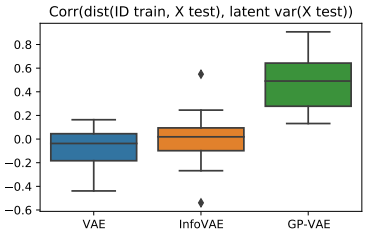


Fig. 4. The Pearson correlation coefficient between the latent variance of each test point (X test) and the average distance between X test and the all data points in the training dataset (ID train).

Metric	log(p(x))			roc				pr			
	VAE	InfoVAE	GP-VAE	VAE	InfoVAE	GP-VAE	Dev-Net	VAE	InfoVAE	GP-VAE	Dev-Net
annthyroid	-47.44	-49.96	**-8.73**	0.64	0.36	**0.94**	0.68	0.32	0.16	**0.67**	0.28
arrhythmia	-306.86	-307.26	**-257.96**	0.62	0.61	**0.81**	0.50	0.55	0.53	**0.69**	0.26
breastw	**-12.73**	-13.06	-13.86	0.44	0.66	**0.99**	0.94	0.62	0.76	**0.99**	0.92
cardio	-28.10	-28.86	**-24.89**	0.31	0.27	**0.98**	0.75	0.13	0.15	**0.93**	0.55
glass	-12.79	**-12.02**	-15.59	0.39	0.61	0.84	**0.86**	0.12	0.17	0.31	**0.32**
ionosphere	-68.74	-75.78	**-36.69**	0.79	0.79	**0.96**	0.81	0.86	0.84	**0.97**	0.80
letter	-98.08	-99.03	**-49.47**	0.60	0.66	**0.85**	0.58	0.25	0.31	**0.49**	0.16
lympho	-37.13	-38.10	**-18.10**	0.55	0.60	**0.89**	0.66	0.34	0.36	**0.66**	0.31
mnist	-132.50	-132.60	**-122.48**	0.59	0.56	**0.96**	0.67	0.40	0.34	**0.86**	0.39
musk	-207.80	-208.09	**-182.41**	0.01	0.01	**1.00**	0.84	0.03	0.03	**1.00**	0.63
optdigits	-92.59	-93.71	**-68.35**	0.61	0.63	0.98	**0.99**	0.09	0.11	0.66	**0.95**
pendigits	**-21.45**	-23.60	-23.22	0.35	0.38	**1.00**	0.93	0.04	0.08	**0.98**	0.79
pima	**-7.53**	-8.12	-13.69	0.47	0.41	**0.70**	0.55	0.52	0.48	**0.69**	0.55
satellite	-167.27	-197.56	**-59.73**	0.48	0.49	**0.80**	0.63	0.53	0.53	**0.86**	0.61
satimage	-155.36	-153.43	**-59.08**	0.30	0.09	**1.00**	0.82	0.02	0.01	**0.98**	0.58
shuttle	**-10.97**	-11.28	-24.38	0.92	0.85	**1.00**	0.97	0.74	0.69	**0.98**	0.95
thyroid	-23.83	-16.54	**-9.18**	0.55	0.71	**0.98**	0.90	0.28	0.49	**0.80**	0.69
vertebral	-9.14	**-8.43**	-9.84	0.38	0.53	0.59	**0.66**	0.20	0.30	0.27	**0.33**
wbc	**-33.07**	-34.25	-45.16	0.29	0.02	**0.97**	0.78	0.25	0.06	**0.84**	0.52
wine	-14.72	**-14.35**	-22.24	0.22	0.34	**0.97**	0.77	0.10	0.22	**0.84**	0.53
FashionMnist	-790.66	-805.52	**-753.12**	0.13	0.23	0.93	**0.97**	0.33	0.37	0.89	**0.97**

Table 1. OOD detection performance of the GP-VAE, a standard VAE, a InfoVAE and deviation network models on the OODS datasets and FashionMnist vs Mnist. We present both roc and pr. Bold values indicate the best performing OOD detection mechanism.

after randomly splitting the ID data into train and test set with five random seeds and report the average performance. The standard deviations across runs are depicted in the Appendix. Compared to supervised OOD methods, we do not actively train the VAE-based models to detect OOD data.

The area under the Receiver Operating Characteristics *roc* and Precision-Recall curve *pr* values are presented in column 4-11 of Table 1. The bold values indicate the best performing models. A general observation is that the GP-VAE outperforms the other VAE-based models on all datasets. This indicates that the GP-VAE produces more reliable latent variance estimates than the standard VAE and InfoVAE. The GP-VAE also outperforms the supervised Dev-Net on 17 out of the 20 OOD datasets. This suggests that our approach is well suited for OOD detection while not requiring labeled training data.

FashionMnist vs Mnist We train each model on the training data of FashionMnist (FM) and test on the test data of FashionMnist and Mnist (M). As before, we compare our approach to a standard VAE, the InfoVAE and Dev-Net. The *roc* and *pr* values are listed in the last row of Table 1. We see that the latent variance of the standard VAE and InfoVAE have no discriminative power. To understand this discrepancy, we depict the VAE's and GP-VAE's latent uncertainty values over the FashionMnist training and testing data and the Mnist testing data in Figure 5. While the OOD data (Mnist) has lower latent uncertainty values than the ID data in the case of the standard VAE, the GP-VAE assigns higher latent uncertainty values to the OOD data compared to the ID data. The extreme behavior of the VAE might be explained by similar arguments as brought forward in [21]; from a

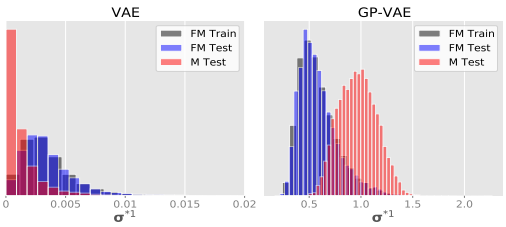


Fig. 5. Histograms of latent uncertainties $\boldsymbol{\sigma}^{*1}$ of the first latent dimension generated by the GP-VAE and the VAE models trained on the FM dataset.

statistical viewpoint the Mnist density lies within the FashionMnist density with lower variance of pixel values. The NN encoder of the VAE might be influenced by these low-level statistics and therefore underestimate the uncertainty over OOD data points.

4.4 OOD Polution of the Training Data

While we investigated whether the latent uncertainty estimates of the GP-VAE can reliably distinguish between ID and OOD data in Section 4.3, in this section we look at how stable the latent uncertainty estimates are in the presence of OOD noise in the ID training data. We therefore analyze how the GP-VAE reacts to different levels of data pollution and compare the behavior to standard VAEs. To this end, we take a look at the Breast Cancer Wisconsin (Original) dataset (breastw) in the ODDS library. The dataset consists of 444 ID data points (split into 222 training and 222 testing points) and 239 OOD data points. Except for introducing OOD noise into the training data, we keep all other settings the same as described in Section 4.3.

We compare the behavior of the GP-VAE to the VAE in Figure 6 by removing 0, 10, 50, 100 and 200 OOD samples from the testing data and adding these to the ID training data. This data split is performed randomly, with ten different seeds over which we average the performance in terms of OOD detection (measured in *roc* and *pr*). The standard deviation between trials is visualized by the colored shadows in Figure 6. We see that the GP-VAE is less susceptible to noise. In addition, the performance of the VAE is random seed dependent, which causes

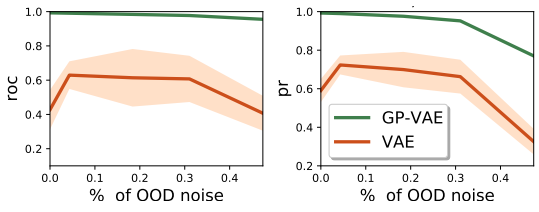
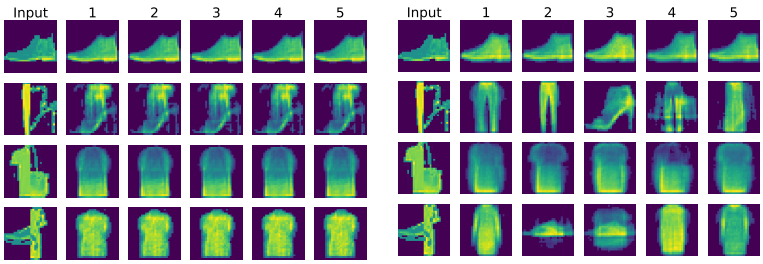


Fig. 6. Performance of the GP-VAE compared to a VAE under the influence of noise in the training data. We present *roc* (left) and *pr* values (right) average and variance over ten random seeds. The between-run variance of the GP-VAE is very small.



(a) VAE (b) GP-VAE

Fig. 7. Decoded samples from the encoded latent variables for the Input image (left row). The top row shows an intact image of the FM dataset. The three bottom rows show OOD images.

the large variations between trials while the across-trial variance of the GP-VAE is too small to be visible in the Figure. This implies that the GP-VAE's reaction to noise is more stable across different noise samples.

4.5 Synthesizing Variants of Input Data

As described in the introduction, VAEs are used to generate different variants of an encoded data point. We would expect the model to generate a larger variety when it is uncertain about the input. To demonstrate how the GP-VAE compares to a VAE in that regard, we use the FM dataset. We generate OOD data points by concatenating two halves of different FM images as shown in the three bottom rows of the *Input* column in Figure 7. We then sample variants of these inputs as shown in columns 1–5 in Figure 7. For comparison, we also sample around an ID image from the FM dataset in the top row. We can see that the VAE fails to sample a larger variety for OOD data as the latent variance is not reliably expressing uncertainty. In contrast, the GP-VAE samples a larger variety and even generates samples from both fashion items that constitute the input image, e.g. a high heel or trousers in the second row.

4.6 Interpretable Kernels

One advantage of GPs is that we are free to choose the kernel function. Building on [5], we use additive kernels of the form $k(\boldsymbol{x}, \boldsymbol{x'}) = \sum_d k^d(\boldsymbol{x}^d, \boldsymbol{x'}^d)$, that is, summing over separate kernels for each feature. This gives rise to interpretable models and automatic relevance determination as each feature dimension is treated independently. We use the same approach to determine which features distinguish ID and OOD data on the *thyroid* dataset. We choose a squared exponential kernel for each input feature dimension and keep the remaining settings the same. To determine how close a data point is to the feature representation learned by the model, we compute the similarity between dimension d of point $\boldsymbol{x}$ and the inducing points $\mathit{similarity}^d = \sum_m k^d(\boldsymbol{x}^d, \boldsymbol{u}_m^d)$. We hypothesize that feature

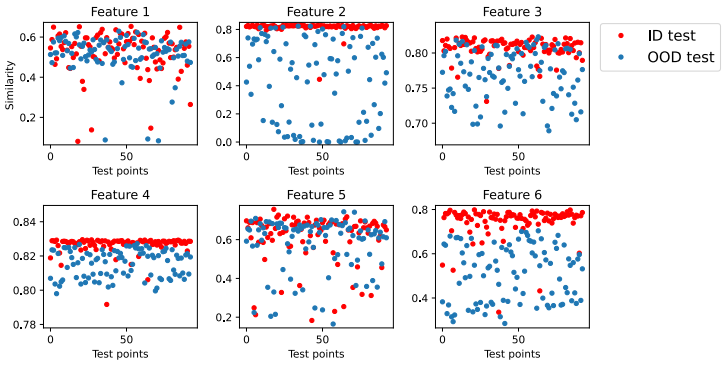


Fig. 8. The similarity$^d = \sum_j k^d(\boldsymbol{x}^d, \hat{\boldsymbol{x}}_j^d)$ score for ID and OOD test data points of the *thyroid* dataset.

dimensions that are important for OOD detection, meaning in which ID and OOD data points differ, will also differ in terms of *similarity*^{d}.

We analyze whether this hypothesis holds for the *thyroid* dataset, which consists of six continuous features describing hormone levels. To determine the importance of each feature to detect OOD, we train a random forest, a decision tree and logistic regression to classify ID and OOD data in a supervised fashion. We normalize the feature importance values determined by these models and average over the three models (see the Appendix for all values). This gives us the feature importance values of $\boldsymbol{x}^1 = 0.01$, $\boldsymbol{x}^2 = 0.39$, $\boldsymbol{x}^3 = 0.11$, $\boldsymbol{x}^4 = 0.19$, $\boldsymbol{x}^5 = 0.04$, $\boldsymbol{x}^6 = 0.26$, which indicates that the hormone features 2 and 6 are important for OOD detection.

When inspecting the similarity scores in Figure 8, it becomes apparent that these features also exhibit abnormal behavior as determined by the additive kernel. The similarity score of the OOD data is lower than for the ID data for these features. To quantify these findings, we fit a Gaussian distribution to the the ID similarity values of each feature and compute the log likelihood of the OOD similarity values under each Gaussian. This gives us the feature importance values of $ll^1 = 83.53$, $ll^2 = $ -6151.94, $ll^3 = 120.36$, $ll^4 = 68.84$, $ll^5 = 49.52$, $ll^6 = $ -488.53, where smaller values indicate higher importance.

By determining whether a set of test points behaves differently in the similarity space of each dimension, we gain a better understanding of which features contribute to OOD detection. This knowledge can be used to make subsequent decisions, e.g. when building a rule-based anomaly detection system.

5 Related Work

A number of works have identified the problem of small, uninformative latent variance values and proposed different ways to overcome them. One solution is to set the latent variance equal to the neural network output plus a small constant value [18]. In a similar manner, the encoded latent variance can be removed from

the model and replaced by a constant that is treated as a hyperparameter [1]. While these approaches achieve the effect that one can sample different variants of an encoded data point, the variance is not reliably representing how uncertain the model is over a given data point. In contrast, our proposed GP encoder learns to express meaningful latent variance estimates for each data point. Another solution to the latent variance problem is to add an additional weighted constraint to the ELBO that keeps the latent variance values from vanishing [23]. However, this additional objective does not ensure reliable uncertainty estimates and comes with the additional burden of having to determine how to optimally weight the additional constraint in the loss function.

As our approach builds on Gaussian processes, we discuss different approaches combining Gaussian processes and VAEs proposed in recent years.

Deep Gaussian Processes have been employed as VAEs, replacing all neural network layers in the decoder network by Gaussian Processes [4] and using neural networks to infer the variational parameters of these GPs. We propose instead to replace the encoder by a GP to infer accurate uncertainty estimates while keeping the neural network structure of the decoder.

Similar to our work, variational Gaussian processes [28] warp samples from a simple distribution with a GP to model the approximate posterior, which in turn is regularized by an auxiliary distribution. Our approach relies on inducing points in the data space and therefore does not require additional auxiliary distributions.

To model correlations in the latent space, several works have introduced a GP prior reflecting structural correlations in the data, see e.g. [2, 11]. It replaces the common choice of an i.i.d. standard normal prior. Since the encoder, or approximate posterior, is still driven by a neural network, obtaining useful latent-space variance estimates remains problematic. In contrast, our work keeps the i.i.d. standard normal prior but replaces the neural network encoder with a sparse GP approximate posterior. It is possible to obtain a model similar to (but distinct from) ours with a structured prior, as discussed in the Appendix.

Finally, [6] employ Gaussian processes for time-series imputation by modeling temporal dependencies in the latent space with a GP. However, the encoding and decoding is performed by neural networks. In contrast, we propose to substitute the encoder with a GP.

6 Conclusion

In this work, we introduced a Gaussian process encoder with sparse inducing points for Variational autoencoders. The combination of Gaussian processes and neural networks, as proposed in this paper, merges the advantages of GPs, such as the ability to encode structure through a kernel function and reliable uncertainty estimates, with the advantages coming with neural networks, such as efficient representation learning and scalability. Our experiments show that GP-VAEs have additional advantages over standard VAEs, such as robustness to noise and the freedom to choose the kernel function.

One disadvantage of GPs compared to neural networks is their limited capability of representing structured, high dimensional data such as images. The kernel function constraints the generalization capability of GPs to local metrics, such as the euclidean distance. However, novel kernel functions, such as convolutional kernels, can be used to operate even in image spaces [29]. In future work we plan to extend our approach to include such priors as well as temporal dynamics and data from multiple sources.

References

1. Braithwaite, D.T., Kleijn, W.B.: Bounded information rate variational autoencoders. KDD Deep Learning Day (2018)
2. Casale, F.P., Dalca, A., Saglietti, L., Listgarten, J., Fusi, N.: Gaussian process prior variational autoencoders. In: Advances in Neural Information Processing Systems. pp. 10369–10380 (2018)
3. Dai, B., Wipf, D.: Diagnosing and enhancing vae models. In: International Conference on Learning Representations (2018)
4. Dai, Z., Damianou, A.C., González, J., Lawrence, N.D.: Variational auto-encoded deep gaussian processes. In: ICLR (Poster) (2016)
5. Duvenaud, D.: Automatic model construction with Gaussian processes. Ph.D. thesis, PhD thesis, University of Cambridge (2014)
6. Fortuin, V., Rätsch, G., Mandt, S.: Multivariate time series imputation with variational autoencoders. In: 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) (2020)
7. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: Advances in neural information processing systems. pp. 2450–2462 (2018)
8. He, J., Spokoyny, D., Neubig, G., Berg-Kirkpatrick, T.: Lagging inference networks and posterior collapse in variational autoencoders. In: International Conference on Learning Representations (2018)
9. Hendrycks, D., Mazeika, M., Dietterich, T.G.: Deep anomaly detection with outlier exposure. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019)
10. Jankowiak, M., Pleiss, G., Gardner, J.: Parametric gaussian process regressors. In: International Conference on Machine Learning. pp. 4702–4712. PMLR (2020)
11. Jazbec, M., Ashman, M., Fortuin, V., Pearce, M., Mandt, S., Rätsch, G.: Scalable gaussian process variational autoencoders. In: International Conference on Artificial Intelligence and Statistics. vol. 130, pp. 3511–3519. PMLR (2021)
12. Johnson, M.J., Duvenaud, D.K., Wiltschko, A., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Advances in neural information processing systems. pp. 2946–2954 (2016)
13. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Foundations and Trends® in Machine Learning **12**(4), 307–392 (2019)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations, ICLR 2014 (2014)
15. scikit learn: Two moons dataset. In: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html, scikit-learn dataset make_moons, accessed 2021 (2021)

16. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)
17. Nalisnick, E.T., Matsukawa, A., Teh, Y.W., Görür, D., Lakshminarayanan, B.: Do deep generative models know what they don't know? In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019)
18. Nash, C., Williams, C.K.: The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In: Computer Graphics Forum. vol. 36, pp. 1–12. Wiley Online Library (2017)
19. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 353–362 (2019)
20. Rayana, S.: Odds library. In: Stony Brook University, Department of Computer Sciences (2016)
21. Ren, J., Liu, P.J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., Lakshminarayanan, B.: Likelihood ratios for out-of-distribution detection. In: Advances in Neural Information Processing Systems. pp. 14680–14691 (2019)
22. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning (2014)
23. Rubenstein, P., Schölkopf, B., Tolstikhin, I.: Learning disentangled representations with wasserstein auto-encoders. In: International Conference on Learning Representations (ICLR 2018) Workshops (2018)
24. Skafte, N., Jørgensen, M., Hauberg, S.: Reliable training and estimation of variance networks. In: Advances in Neural Information Processing Systems. pp. 6326–6336 (2019)
25. Snelson, E., Ghahramani, Z.: Local and global sparse gaussian process approximations. In: Artificial Intelligence and Statistics. pp. 524–531 (2007)
26. Titsias, M.: Variational learning of inducing variables in sparse gaussian processes. In: Artificial Intelligence and Statistics. pp. 567–574 (2009)
27. Tomczak, J., Welling, M.: Vae with a vampprior. In: International Conference on Artificial Intelligence and Statistics. pp. 1214–1223 (2018)
28. Tran, D., Ranganath, R., Blei, D.M.: The variational gaussian process. In: 4th International Conference on Learning Representations, ICLR 2016 (2016)
29. Van der Wilk, M., Rasmussen, C.E., Hensman, J.: Convolutional gaussian processes. In: Advances in Neural Information Processing Systems. pp. 2849–2858 (2017)
30. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
31. Zhang, C., Bütepage, J., Kjellström, H., Mandt, S.: Advances in variational inference. IEEE transactions on pattern analysis and machine intelligence **41**(8), 2008–2026 (2018)
32. Zhao, S., Song, J., Ermon, S.: Infovae: Balancing learning and inference in variational autoencoders. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5885–5892 (2019)