EasyChair Preprint

№ 4448

Uncertainty Estimates in Deep Generative Models using Gaussian Processes

Kai Katsumata and Ryoga Kobayashi

October 20, 2020

Uncertainty Estimates in Deep Generative Models using Gaussian Processes

Kai Katsumata[(✉)1][0000−0001−9729−2588] and Ryoga Kobayashi[2][0000−0003−0408−1891]

[1] Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan
katsumata@nlab.ci.i.u-tokyo.ac.jp
[2] Graduate School of Media and Governance, Keio University, Tokyo, Japan
ryoga@sfc.keio.ac.jp

Abstract. We propose a new framework to estimate the uncertainty of deep generative models. In real-world applications, uncertainty allows us to evaluate the reliability of the outcome of machine learning systems. Gaussian processes are widely known as a method in machine learning which provides estimates of uncertainty. Moreover, Gaussian processes have been shown to be equivalent to deep neural networks with infinitely wide layers. This equivalence suggests that Gaussian process regression can be used to perform Bayesian prediction with deep neural networks. However, existing Bayesian treatments of neural networks via Gaussian processes have only been applied so far to supervised learning; we are not aware of any work using neural networks and Gaussian processes for unsupervised learning. We extend the Bayesian Gaussian process latent variable model, an unsupervised learning method using Gaussian processes, and propose a Bayesian deep generative model by approximating the expectations of complex kernels. With a series of experiments, we validate that our method provides estimates of uncertainty from the relevance between variance and the output quality.

Keywords: Gaussian process · Neural network · Deep learning · Gaussian process latent variable model · Bayesian learning

1 Introduction

In this paper, we propose a Bayesian deep generative model using Gaussian process latent variable models (GPLVMs), which calculate predictive distributions to treat the uncertainty. There has been great discussion about the reliability of deep learning models. Especially, a lack of reliability is a big issue with deep neural networks [12]. This weakness is due to the fact that fitting performance is the primary focus of deep models, and reliability is often overlooked [22]. We tackle the issue by using Gaussian processes, which offer reliable posterior predictive distributions from a Bayesian perspective [21]. Some studies [9, 23] claim that many existing deep learning models that only predict the expected value

of outputs sometimes provide unreliable outcomes. This behavior is hypothesized to be caused by the use of a deterministic inference process and focusing on model fitting [6]. Then, the current deep neural networks cannot be fully trusted because they work well only in unrealistically idealized environments, and provide unreliable outputs for unforeseen inputs in the real-world.

Lee et al. [16] approach the reliability issue of deep neural networks by constructing Bayesian neural networks from Gaussian processes in the context of supervised learning. Gaussian processes that correspond to a deep neural network achieve high performance and provide estimates of uncertainty. Bayesian GPLVMs [24], which use Gaussian processes for unsupervised learning, can only be used with a few kernels as the model requires analytic formulas involving the kernel function for optimization.

In this paper, we combine Bayesian GPLVMs and deep kernels for reliable deep generative models. We cannot straightforwardly combine these methods, because a close form solution for the ψ statistics cannot be derived. Employing Monte-Carlo approximation allows us to combine Bayesian GPLVMs and deep kernels because it provides differentiable methods for calculating the ψ statistics. The model can exploit state-of-the-art DNN architectures using deep kernels corresponding to these architectures and suit various datasets. Our work differs from many existing deep generative models [9, 14] in that our method can control the quality of output because it can treat the uncertainty. In the experiments, we remove unconfident data points to reduce the gap between the actual data distribution and the generated data distribution. Our contributions are as follows:

- We propose a GPLVM based deep generative model by incorporating deep kernels into Bayesian GPLVMs.
- We demonstrate that our method provides a useful estimate of uncertainty through experiments that investigate the relevance between the variance and the quality of the expectations in the vein of Lee et al. [16].

2 Related Work

The majority of the research into uncertainty of deep learning focuses on supervised learning. Neal [18] has shown that infinitely wide neural networks are equivalent to Gaussian processes. Moreover, Lee et al. [16] have constructed infinitely wide neural networks using Gaussian processes and have achieved higher prediction accuracy than neural networks trained via gradient methods. Recent work extends the framework introduced by Lee et al. [16] for fully-connected neural networks to other deep neural network architectures. Novak et al. [19] proved the correspondence between deep convolutional neural networks and Gaussian processes, and Garriga-Alonso et al. [7] have shown the correspondence between deep convolutional neural networks with residual blocks and Gaussian processes. Yang [25] unified these results by introducing a notation for expressing various neural network layers and revealed relationships between Gaussian processes and various architectures of neural networks.

An area that is less related is that of Deep Gaussian Processes [5, 2]. They are concerned with stacking Gaussian processes to construct rich models. Our study differs from these in that our model corresponds to a deep neural network. They also employ the reparametrization trick to propagate gradients between Gaussian process components rather than for Monte-Carlo integration.

On the other hand, little attention has been given to the relationship between Gaussian processes and deep generative models for unsupervised learning. We find clues in GPLVMs [15, 24] to answer this question. Lawrence [15] converted Gaussian process regression to latent variable model called GPLVMs, which models potentially nonlinear relationships between observed data and latent variables using a Gaussian process. GPLVMs require complex computation for optimization, unlike regular Gaussian process regression.

3 Bayesian GPLVM

In this section, we introduce Bayesian GPLVMs. Let $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ be observed data where N is the number of observations, and D is the dimension of a data point. Latent variables $\boldsymbol{X} \in \mathbb{R}^{N \times Q}$ are not observed where Q is the dimension of a latent point. We can express the likelihood function for a data point under a Gaussian process as

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{y}_d|\boldsymbol{0}_N, \boldsymbol{K}_{\boldsymbol{XX}} + \beta^{-1}\boldsymbol{I}_N).$$

where $\boldsymbol{y}_d$ is the d-th column of $\boldsymbol{Y}$, the kernel matrix $\boldsymbol{K}_{\boldsymbol{XX}}$ is an $N \times N$ covariance matrix defined by a kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ such as linear kernels, and β is a hyperparameter corresponding to the precision of the additive Gaussian noise. The mapping from latent variable space to observed data space is performed via Gaussian process regression. It is possible to interpret the generation process of Bayesian GPLVMs as deep neural networks that map unobserved latent variables to observed data. By using this variational distribution q with variational parameters $\{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}_{n=1}^N$, Jensen's lower bound on log marginal likelihood $\log p(\boldsymbol{Y})$ can be expressed as:

$$\begin{aligned} F(q) &= \int q(\boldsymbol{X}) \log p(\boldsymbol{Y}|\boldsymbol{X}) d\boldsymbol{X} - \int q(\boldsymbol{X}) \log \frac{q(\boldsymbol{X})}{p(\boldsymbol{X})} d\boldsymbol{X} \\ &= \sum_{d=1}^{D} \tilde{F}_d(q) - \mathrm{KL}(q\|p). \end{aligned}$$

Titsias and Lawrence [24] have given $\tilde{F}_d(q)$ for optimization of Bayesian GPLVMs. Using the set of inducing points $\boldsymbol{Z} \in \mathbb{R}^{M \times Q}$ and Ψ statistics: $\psi_0 = \mathrm{tr}\left(\langle \boldsymbol{K}_{\boldsymbol{XX}} \rangle_{q(\boldsymbol{X})}\right)$, $\boldsymbol{\Psi}_1 = \langle \boldsymbol{K}_{\boldsymbol{XZ}} \rangle_{q(\boldsymbol{X})}$, and $\boldsymbol{\Psi}_2 = \langle \boldsymbol{K}_{\boldsymbol{ZX}} \boldsymbol{K}_{\boldsymbol{XZ}} \rangle_{q(\boldsymbol{X})}$, the closed-form evidence lower bound is

$$\tilde{F}_d(q) \geq \log \left[\frac{\beta^{\frac{N}{2}} |\boldsymbol{K}_{\boldsymbol{ZZ}}|^{\frac{1}{2}}}{(2\pi)^{\frac{N}{2}} |\beta \boldsymbol{\Psi}_2 + \boldsymbol{K}_{\boldsymbol{ZZ}}|^{\frac{1}{2}}} \exp(-\frac{1}{2} \boldsymbol{y}_d^T W \boldsymbol{y}_d) \right] - \frac{\beta \psi_0}{2} + \frac{\beta}{2} \mathrm{tr}(\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1} \boldsymbol{\Psi}_2),$$

where $W = \beta \boldsymbol{I}_N - \beta^2 \boldsymbol{\Psi}_1 (\beta \boldsymbol{\Psi}_2 + \boldsymbol{K}_{\boldsymbol{ZZ}})^{-1} \boldsymbol{\Psi}_1^{\boldsymbol{T}}$ and, we define expectations under the distribution $q(\boldsymbol{X})$ as $\langle \cdot \rangle_{q(\boldsymbol{X})}$. The rest of $F(q)$ consists of the Kullback Leibler divergence between two Gaussian distributions and can be analytically derived. As a result of this approximation, we can avoid calculating the $N \times N$ covariance matrix in the optimization process. We optimize the variational parameters $\{\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2\}_{n=1}^N$ and $\boldsymbol{Z}$ with gradient-based methods such as Adam [13]. Using integral notation, ψ_0 is written as

$$\psi_0 = \sum_{n=1}^{N} \int k(\boldsymbol{x}_n, \boldsymbol{x}_n) \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) d\boldsymbol{x}_n.$$

$\boldsymbol{\Psi}_1$ is the $N \times M$ matrix such that

$$(\boldsymbol{\Psi}_1)_{nm} = \int k(\boldsymbol{x}_n, \boldsymbol{z}_m) \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) d\boldsymbol{x}_n.$$

Here, $\boldsymbol{z}_m$ is the m-th row of $\boldsymbol{Z}$. $\boldsymbol{\Psi}_2$ is the $M \times M$ matrix such that

$$(\boldsymbol{\Psi}_2)_{mm} = \sum_{n=1}^{N} \int k(\boldsymbol{x}_n, \boldsymbol{z}_m) k(\boldsymbol{z}'_m, \boldsymbol{x}_n) \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2) d\boldsymbol{x}_n.$$

However, we only obtain analytic forms of the Ψ statistics for a few simple kernels. We construct Bayesian deep generative models by applying deep kernels [4] to Bayesian GPLVMs in the manner of Lee et al. [16]. In Bayesian GPLVMs, the distribution of observed data follows Gaussian processes. Since this distribution corresponds to the distribution in a Gaussian process, the Gaussian process followed by $\boldsymbol{y}_d$ given the latent variable $\boldsymbol{X}$ in GPLVMs is equivalent to neural networks. The decoder of Bayesian GPLVMs fixed latent variables is equivalent to Gaussian process regression.

4 Approximate $\boldsymbol{\Psi}$ Statistics

In this section, we aim to integrate deep kernels into Bayesian GPLVMs. We introduce an approximation of Ψ statistics by employing Monte-Carlo integration to intractable deep kernels. Titsias and Lawrence [24] have only shown the analytic solution for simple kernel functions such as RBF kernels and linear kernels. However, their analytical solution is not applicable to other types of kernels, including deep kernels. Deep kernels have high capacity as these are recursive and especially complex. In fact, previous studies achieve high performance on image classification tasks by using Gaussian process regression involving these kernels. As the integral is intractable, we perform a differentiable approximate integration to derive the required Ψ statistics. The expected value of $f(\boldsymbol{x})$ following a distribution $p(\boldsymbol{x})$ can be approximated by $\hat{f} = \frac{1}{\tau} \sum_{i=1}^{\tau} f(\boldsymbol{x}_i)$ where $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\tau$ are i.i.d samples drawn from $p(\boldsymbol{x})$, and τ is the total number of samples. This is an unbiased estimator since $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$. The approximation has

the variance $\mathrm{Var}[\hat{f}] = \frac{1}{\tau}\mathbb{E}\left[(f - \mathbb{E}[f])^2\right]$. The accuracy of the estimator depends only on the number of sampling points and not on the dimensionality of $\boldsymbol{x}$. We can safely apply this to high dimensional latent variables.

Assume that $\boldsymbol{x}$ follows a Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$. We want to sample data from the distribution $p(\boldsymbol{x})$, but sampling is generally non-differentiable. We use the reparameterization trick

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_D, \boldsymbol{I}_D), \quad \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \tag{1}$$

as used in variational autoencders (VAEs) for differentiable sampling [14] from Gaussian distributions. Using Monte-Carlo approximation and Eq. (1), we can calculate the Ψ statistics as

$$\psi_0 \approx \frac{1}{\tau}\sum_{n=1}^{N}\sum_{i=1}^{\tau} k(\boldsymbol{x}_{ni}, \boldsymbol{x}_{ni}), \tag{2}$$

$$(\boldsymbol{\Psi}_1)_{nm} \approx \frac{1}{\tau}\sum_{i=1}^{\tau} k(\boldsymbol{x}_{ni}, \boldsymbol{z}_m), \tag{3}$$

$$(\boldsymbol{\Psi}_2)_{mm'} \approx \frac{1}{\tau}\sum_{n=1}^{N}\sum_{i=1}^{\tau} k(\boldsymbol{x}_{ni}, \boldsymbol{z}_m) k(\boldsymbol{z}'_m, \boldsymbol{x}_{ni}), \tag{4}$$

respectively, where $\boldsymbol{x}_{ni} = \boldsymbol{\mu}_n + \boldsymbol{\sigma}_n \cdot \boldsymbol{\epsilon}_i$. The inducing point $\boldsymbol{z}_m$ is a variational parameter but is treated as a constant in the calculations of expected values due to the integration of $\boldsymbol{x}_n$. The lower bound can be jointly maximized over the variational parameters $\{(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n^2)\}_{n=1}^N$ and $\boldsymbol{Z}$ given Eqs. (2) to (4) for deep kernels using gradient methods.

Using optimized variational parameters consisting of $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and $\boldsymbol{Z}$, we can obtain the mean and variance given new latent points, which based on the results of Quiñonero-Candela et al. [20], take the form:

$$\mathbb{E}\left[\boldsymbol{y}^* | \boldsymbol{x}^*, \mathcal{D}\right] = B^{\boldsymbol{T}} \boldsymbol{\Psi}_1^*, \tag{5}$$

$$\begin{aligned}\mathrm{Var}\left[\boldsymbol{y}^* | \boldsymbol{x}^*, \mathcal{D}\right] =& B^{\boldsymbol{T}}(\boldsymbol{\Psi}_2^* - \boldsymbol{\Psi}_1^*(\boldsymbol{\Psi}_1^*)^{\boldsymbol{T}})B - \mathrm{tr}\left(\left[\boldsymbol{K}_{MM}^{-1} - (\boldsymbol{K}_{MM} + \beta\boldsymbol{\Psi}_2)^{-1}\right]\boldsymbol{\Psi}_2^*\right)\boldsymbol{I}_D \\ &+ \psi_0^* \boldsymbol{I}_D + \beta^{-1}\boldsymbol{I}_D.\end{aligned} \tag{6}$$

Here, $B = \beta\left(\boldsymbol{K}_{MM} + \beta\boldsymbol{\Psi}_2\right)^{-1}\boldsymbol{\Psi}_1^{\boldsymbol{T}}\boldsymbol{Y}$, $\psi_0^* = \mathrm{tr}\left(\langle \boldsymbol{K}_{\boldsymbol{x}^*\boldsymbol{x}^*}\rangle_{q(\boldsymbol{x}^*)}\right)$, $\boldsymbol{\Psi}_1^* = \langle \boldsymbol{K}_{\boldsymbol{Z}\boldsymbol{x}^*}\rangle_{q(\boldsymbol{x}^*)}$, and $\boldsymbol{\Psi}_2^* = \langle \boldsymbol{K}_{\boldsymbol{Z}\boldsymbol{x}^*}\boldsymbol{K}_{\boldsymbol{x}^*\boldsymbol{Z}}\rangle_{q(\boldsymbol{x}^*)}$. $\boldsymbol{x}^*$ consists of $\{\boldsymbol{\mu}^*, \boldsymbol{\sigma}^{2*}\}$, and $\mathcal{D}$ consists of training data $\boldsymbol{Y}$ and optimized variational parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and $\boldsymbol{Z}$.

We obtain the fixed latent variables after the optimization. We also earn the expected value and variance for new latent space points employing Eq. (5) and Eq. (6) on the optimized latent variables just like the Bayesian GPLVM. Those results can be applied to reliable classification on the latent space and reliable data generation.

Table 1. The average of the classification accuracy of the nearest neighbor classification in latent space constructed by each model. We report the averaged accuracy on the test dataset over five trials. For each trial, we use 80% of the dataset as the training set and the rest as the test set. Our method achieves superior performance over other methods in the USPS dataset.

	Oil Flow	USPS
VAE [14]	89.0	87.1
Bayesian GPLVM (Linear) [24]	89.0	79.8
Bayesian GPLVM (RBF) [24]	**96.0**	90.4
Our method	95.0	**93.9**

5 Experiments

To evaluate the proposed method, we now conduct experiments with some standard machine learning datasets and image datasets. Some experiments report the classification accuracy on learned latent representations and the quality of reconstructed data. The last two experiments in Section 5.3 aim to validate the usefulness of predicted uncertainty, the principal purpose of this study. In Section 5.3, we verify the usefulness of the variance on the latent space and demonstrate the usefulness of the variance of the generated data space.

Our models are composed of four layers for all experiments in this section. The number of samples used to approximate Ψ statistics is set to 10. In our model and the Bayesian GPLVM, the means in the variational distribution are initialized based on PCA, the variances of the variational distribution are initialized to 0.1, and 20 inducing points are used. We use Adam [13] as an optimization method for all models used in our experiments.

5.1 Classification Experiments

We apply the method to multi-phase oil flow data [1] and compare classification accuracy with related work, just like experiments in Titsias and Lawrence [24]. The dataset consists of 1000 observations distributed equally among the three classes. Each 12-dimensional data point belongs to one of three different geometrical configurations. The first experiment aims to show that our model learns better latent representations than existing methods. We take the latent variables to be ten-dimensional. For comparison, Table 1 shows the average and of the accuracy of k-nearest neighbor method (k=5) trained on the latent space over five runs for our model, the Bayesian GPLVM, and the VAE. We apply a linear kernel and an RBF kernel to the Bayesian GPLVM for the experiments. First, we mapped all 1000 of the data points to latent space using each model. Second, we considered the nearest neighbor classifier in the latent space to quantify the quality of latent representations optimized by each model. The encoder and decoder of VAE are composed of four layers for a fair comparison with our model. The dimension of the hidden layer in the VAE is hand-tuned between

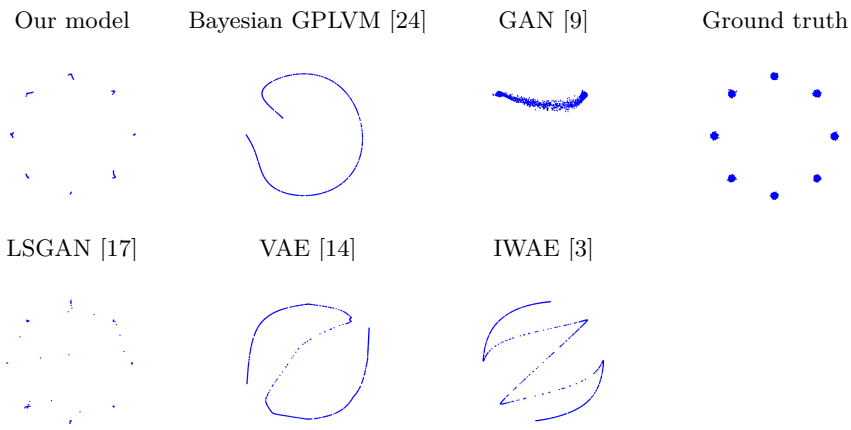

Fig. 1. Generated data mapped from latent space sampled randomly. The first row shows results from the proposed method, Bayesian GPLVM, GAN and the ground truth data points. The second row shows results from the LSGAN, VAE, and IWAE. Our method produces the closest distribution to the ground truth distribution.

32 and 512 to achieve the best accuracy. The accuracy of our model was five points higher than the Bayesian GPLVM with linear kernels and the VAE and was slightly worse than the Bayesian GPLVM with RBF kernels.

We also illustrate the method in handwritten digit recognition datasets as with an above experiment. We conduct the nearest neighbor classification in the latent space for the subset of 7291 of the digits 0-9 from the USPS dataset [10]. Each image of size 16×16 pixels was transformed into the vector row of dimension 256, representing handwritten digits as inputs to a fully connected layer. Table 1 shows results by our model, the Bayesian GPLVM with linear kernels, the Bayesian GPLVM with RBF kernels, and the VAE. For all models, we use ten latent dimensions. We use those models with the same setting as the experiments for Oil Flow data. We report the averaged accuracy over five independent runs for each model. Our model achieves the highest accuracy in the four models, including deterministic deep generative models and classical generative models.

5.2 Gaussian Mixture Distribution Data

Fig. 1 shows the results of the data generation experiment on the toy dataset consisting of samples from Gaussian mixture distribution. The distribution has eight modes, which are arranged in a circle. The dataset consists of 100 data points drawn with equal probability from the eight Gaussian distributions with different means. This experiment aims to show whether the model can reproduce this distribution from the training data. We compare the proposed method with the Bayesian GPLVM, Generative Adversarial Network (GAN) [8], LSGAN [17], VAE, and imporatance weighted autoencders (IWAE) [3]. The LSGAN and IWAE are improved models of the GAN and VAE, respectively. The

Table 2. The negative log likelihood between the generated data and the ground truth data for experiments on grayscale and RGB images datasets and the classification accuracy. Our method outperforms other methods on almost datasets in both classification accuracy and negative log likelihood.

Method	MNIST		Fashion-MNIST		CIFAR-10		CIFAR-100	
	NLL ↓	Acc ↑	NLL ↓	Acc ↑	NLL ↓	Acc ↑	NLL ↓	Acc ↑
VAE [14]	125.32	76.7	247.65	61.6	1941.77	24.4	1899.36	4.5
IWAE [3]	126.23	87.6	**241.77**	69.2	1927.23	21.6	1927.23	4.6
Bayesian GPLVM [24]	122.01	87.7	246.98	74.1	1887.05	25.7	1829.82	5.4
Our method	**121.80**	**87.8**	244.87	**78.1**	**1860.27**	**27.4**	**1809.17**	**6.3**

generated data is computed by $\mathbb{E}[\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}]$, where $\mathcal{D}$ is the training data following the two-dimensional Gaussian mixture distribution and the variational parameters, and $\boldsymbol{x}^*$ is the test data points sampled from a uniform distribution in all models. All the models without Bayesian GPLVM are composed of four layers. We take the latent variables for all models to be one-dimensional as the dataset consists of two-dimensional data. Our model requires the latent dimension less than the data dimension, and we need the same condition for both models.

Fig. 1 illustrates the results of this experiment. In Fig. 1, the rightmost panel shows a scatter plot of the ground truth data, which is the mixture of eight Gaussian distributions. The GAN only generates two out of the eight modes of Gaussian mixture distribution. The phenomenon is referred to as the mode collapse and lack of diversity these are some of the most common issues with vanilla GANs [11]. In contrast, our model successfully generates all eight modes of the Gaussian mixture distribution. We conduct a quantitative evaluation using Maximum Mean Discrepancy (MMD). The MMD scores of our model, Bayesian GPLVM, GAN, LSGAN, VAE, and IWAE are 0.0335, 0.1814, 1.0975, 0.0571, 0.3228, and 0.1644, respectively. The generated distribution by our model is closest to the ground truth distribution in all models.

Furthermore, we conduct experiments for additional image datasets, MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100. We pick 20 or 200 samples per class for each dataset. MNIST and Fashion-MNIST contain 28×28 grayscale images, and CIFAR-10 and CIFAR-100 contain 32×32 RGB images. We compare the proposed method with the VAE, IWAE, and Bayesian GPLVM. Table 2 shows the classification accuracy in the manner described above and the negative log likelihood of reconstructed data from latent points. The whole dataset size is 2000 for all datasets. The proposed method outperforms other methods, without the negative log likelihood on Fashion-MNIST.

5.3 Variance Analysis

The remaining experiments demonstrate the usefulness of uncertainty of outputs obtained by our model. First, we show that the model allocates low variance to

Table 3. Comparative study of the averaged variance of correct and incorrect predictions. The left column shows Eq. (7) and the right column shows Eq. (8). Our method allocates the low variance to high confidence outputs and the high variance to low confidence outputs, unlike the VAE.

	correct	incorrect
VAE	14.911	14.357
Our model	4.277	16.859

confident outputs and allocates high variance to unconfident outputs. Second, we disregard generated samples with a large variance to reduce the gap between the true data distribution and the generated data distribution by our models.

We compare the variance of outputs between our method and VAE following the variance analysis in Lee et al. [16]. The purpose of this experiment is to show whether the variance of the latent variables obtained through the optimization of the model is useful for solving classification problems in the latent variable space. We compare the variance of correct and incorrect classification results for test data by performing nearest neighbors over expectations $\boldsymbol{\mu}$ in the latent space constructed by each model. We agree that $\boldsymbol{\mu}^*$ and $\boldsymbol{\mu}'$ denote a set of size N of latent variables randomly sampled from $\boldsymbol{\mu}$ and its complement set $\boldsymbol{\mu} \setminus \boldsymbol{\mu}^*$. The predicted label corresponding to $\boldsymbol{\mu}_n$ with nearest neighbors with $\boldsymbol{\mu}'$ is defined by $knn_{\boldsymbol{\mu}'}(\boldsymbol{\mu}_n)$. Let the average variance of correct data be defined as

$$\frac{\sum_{n=1}^{N}\sum_{i=1}^{Q}\boldsymbol{\sigma}_{n,i}^2\mathbf{1}\left(knn_{\boldsymbol{\mu}'}(\boldsymbol{\mu}_n^*)=\boldsymbol{y}_n\right)}{\sum_{n=1}^{N}\mathbf{1}\left(knn_{\boldsymbol{\mu}'}(\boldsymbol{\mu}_n^*)=\boldsymbol{y}_n\}\right)}, \tag{7}$$

and let the average variance of incorrect data be defined as

$$\frac{\sum_{n=1}^{N}\sum_{i=1}^{Q}\boldsymbol{\sigma}_{n,i}^2\mathbf{1}\left(knn_{\boldsymbol{\mu}'}(\boldsymbol{\mu}_n^*)\neq\boldsymbol{y}_n\right)}{\sum_{n=1}^{N}\mathbf{1}\left(knn_{\boldsymbol{\mu}'}(\boldsymbol{\mu}_n^*)\neq\boldsymbol{y}_n\}\right)}, \tag{8}$$

where $\boldsymbol{\mu}_n^*$ is a test point in $\boldsymbol{\mu}^*$, $\boldsymbol{\sigma}_n^2$ is the variance of latent variables and corresponds to $\boldsymbol{\mu}_n^*$, and $\boldsymbol{y}_n$ is a given label corresponding to $\boldsymbol{\mu}_n$.

Table 3 shows the averaged variance on correct and incorrect predictions for the VAE and our model. For the VAE, the averaged variance on incorrect predictions is approximately equal to that on correct predictions. However, the averaged variance on incorrect predictions in our model is four times larger than that on correct predictions. In other words, our model allocates large variance for untrust outputs corresponding to unpredictable inputs. In our model, a large variance implies a high uncertainty of prediction and a small variance implies a low degree of uncertainty.

Figs. 2 and 3 illustrate the relationship between the variance of outputs and the quality of the generated data. Fig. 2 shows the generated data reconstructed from the latent variables independently sampled from a uniform distribution over [-3, 3] with our model, and Fig. 3 also shows the generated data with the

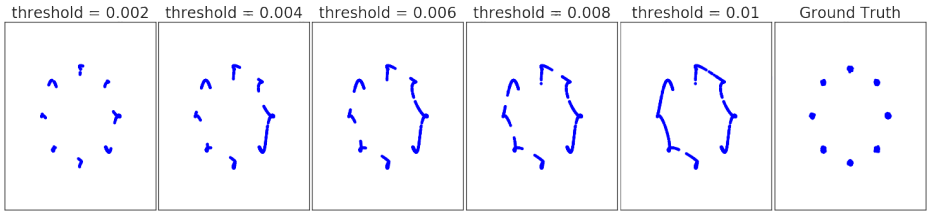

Fig. 2. Comparative study of distributions generated for each threshold by our model. Each column shows a scatter plot of generated samples with the variance less than the threshold. The rightmost column shows a scatter plot of the training data. Our method controls the quality of generated samples using the variance of predictions.

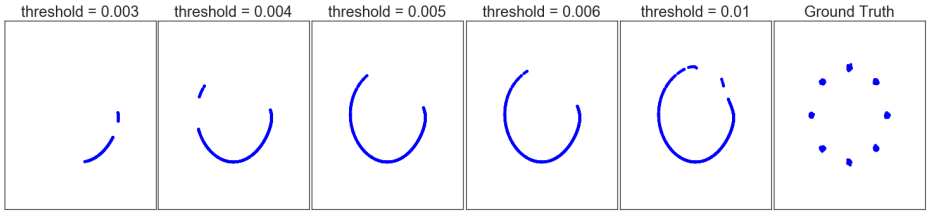

Fig. 3. Comparative study of distributions generated for each threshold of the Bayesian GPLVM with an RBF kernel.

Bayesian GPLVM model. The purpose of this experiment is to show the relevance between the variance and quality of the generated data by the optimized model. We exclude data points with variance greater than or equal to the threshold in each panel. We plot the set of points:

$$\left\{\boldsymbol{m}_n | \max(\boldsymbol{s}_n^2) < t\right\}, \quad n = 1, 2, \ldots, N,$$

where $\boldsymbol{m}_n$ is $\mathbb{E}\left[\boldsymbol{y}_n^* | \boldsymbol{x}_n^*, \mathcal{D}\right]$, $\boldsymbol{s}_n^2$ is $\mathrm{Var}\left[\boldsymbol{y}_n^* | \boldsymbol{x}_n^*, \mathcal{D}\right]$, N is the size of test points, and t is a threshold. The variational parameters $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, and $\boldsymbol{Z}$ of $\mathcal{D}$ are learned latent variables from the training dataset by each model. We note that these thresholds depend on the dataset and network architecture.

In Figs. 2 and 3, we show five plots with different thresholds and the ground truth plot. In Fig. 2, the plot most similar to the ground truth plot is the leftmost plot with the smallest threshold. Conversely, when we assign a large threshold, the model generates a different distribution from the ground truth data distribution. Our model produces a closer distribution to the ground truth distribution with smaller thresholds, unlike Bayesian GPLVM. From the above results, our model control the quality of generated data using the confidence of predictions.

6 Conclusion

In this paper, we develop a Bayesian deep generative model that produces an estimate of uncertainty for generated data by applying deep kernels to Bayesian GPLVMs. For this purpose, we employ an approximate intractable integration to evaluate expectations of deep kernel functions. We present a series of experiments showing that the proposed method offers uncertainty of model outputs, which can then be used for decision-making at higher levels and post-processes. Our model has an advantage compared to the deep generative model and the classical Bayesian generative model. Moreover, we show that our models can provide a useful estimate of uncertainty based on the comparison of the variance of credible predictions and incredible predictions.

Acknowledgement. The authors have benefited from many suggestions and English editing from Mayayuki Takeda and Ryo Kamoi.

References

1. Bishop, C.: Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. Nuclear Instruments and Methods in Physics Research **A327**, 580–593 (1993), https://doi.org/10.1016/0168-9002(93)90728-Z
2. Bui, T.D., Hernández-Lobato, J.M., Hernández-Lobato, D., Li, Y., Turner, R.E.: Deep gaussian processes for regression using approximate expectation propagation. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, ICML'16, vol. 48, pp. 1472–1481 (2016)
3. Burda, Y., Grosse, R.B., Salakhutdinov, R.: Importance weighted autoencoders. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR (2016)
4. Cho, Y., Saul, L.K.: Kernel methods for deep learning. In: Conference on Neural Information Processing Systems (NIPS) (2009)
5. Damianou, A., Lawrence, N.: Deep Gaussian processes. In: Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS), pp. 207–215, AISTATS'13 (2013)
6. Gal, Y.: Uncertainty in Deep Learning. Ph.D. thesis, University of Cambridge (2016)
7. Garriga-Alonso, A., Rasmussen, C.E., Aitchison, L.: Deep convolutional networks as shallow gaussian processes. In: International Conference on Learning Representations (ICLR) (2019)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Conference on Neural Information Processing Systems (NIPS) (2014)
9. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)

10. Hull, J.J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(5), 550–554 (1994), https://doi.org/10.1109/34.291440
11. Im, D.J., Ma, A.H., Taylor, G.W., Branson, K.: Quantitatively evaluating GANs with divergences proposed for training. In: International Conference on Learning Representations (ICLR) (2018)
12. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Conference on Neural Information Processing Systems (NIPS) (2017)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: nternational Conference on Learning Representations (ICLR) (2015)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (ICLR) (2014)
15. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Conference on Neural Information Processing Systems (NIPS) (2003)
16. Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., Bahri, Y.: Deep neural networks as gaussian processes. In: International Conference on Learning Representations (ICLR) (2018)
17. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z.: Multi-class generative adversarial networks with the L2 loss function. CoRR **abs/1611.04076** (2016)
18. Neal, R.M.: Bayesian Learning for Neural Networks. Ph.D. thesis, University of Toronto (1995)
19. Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D.A., Pennington, J., Sohl-Dickstein, J.: Bayesian deep convolutional networks with many channels are gaussian processes. In: International Conference on Learning Representations (ICLR) (2019)
20. Quiñonero-Candela, J., Girard, A., Rasmussen, C.E.: Prediction at an uncertain input for gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting. Tech. Rep. IMM-2003-18 (2003)
21. Rasmussen, C.E., Williams, C.K.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, MIT Press (2006)
22. Serre, T.: Deep learning: The good, the bad, and the ugly. Annual Review of Vision Science **5**(1), 399–426 (2019), https://doi.org/10.1146/annurev-vision-091718-014951
23. Tabacof, P., Tavares, J., Valle, E.: Adversarial images for variational autoencoders **abs/1612.00155** (2016)
24. Titsias, M., Lawrence, N.D.: Bayesian gaussian process latent variable model. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2010)
25. Yang, G.: Tensor programs I: Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In: Conference on Neural Information Processing Systems (NeurIPS) (2019)