

VAE-KRNET AND ITS APPLICATIONS TO VARIATIONAL BAYES

XIAOLIANG WAN AND SHUANGQING WEI

ABSTRACT. In this work, we have proposed a generative model, called VAE-KRnet, for density estimation or approximation, which combines the canonical variational autoencoder (VAE) with our recently developed flow-based generative model, called KRnet. VAE is used as a dimension reduction technique to capture the latent space, and KRnet is used to model the distribution of the latent variable. Using a linear model between the data and the latent variable, we show that VAE-KRnet can be more effective and robust than the canonical VAE. VAE-KRnet can be used as a density model to approximate either data distribution or an arbitrary probability density function (PDF) known up to a constant. VAE-KRnet is flexible in terms of dimensionality. When the number of dimensions is relatively small, KRnet can effectively approximate the distribution in terms of the original random variable. For high-dimensional cases, we may use VAE-KRnet to incorporate dimension reduction. One important application of VAE-KRnet is the variational Bayes for the approximation of the posterior distribution. The variational Bayes approaches are usually based on the minimization of the Kullback-Leibler (KL) divergence between the model and the posterior. For high-dimensional distributions, it is very challenging to construct an accurate density model due to the curse of dimensionality, where extra assumptions are often introduced for efficiency. For instance, the classical mean-field approach assumes mutual independence between dimensions, which often yields an underestimated variance due to oversimplification. To alleviate this issue, we include into the loss the maximization of the mutual information between the latent random variable and the original random variable, which helps keep more information from the region of low density such that the estimation of variance is improved. Numerical experiments have been presented to demonstrate the effectiveness of our model.

1. INTRODUCTION

The density estimation of high-dimensional data plays an important role in unsupervised learning, which is challenging due to the curse of dimensionality [27]. In the last decade, deep generative modeling has made a lot of progress by incorporating with deep neural networks. Deep generative models are usually with likelihood-based methods, such as the autoregressive models [14, 22, 23, 24], variational autoencoders (VAE) [18, 21, 16], and flow-based generative models [7, 25, 8, 19, 33, 3, 6, 9]. One flexible model that does not need the likelihood is the generative adversarial network (GAN) [13, 1], which seeks a Nash equilibrium of a zero-sum game between the generator and the discriminator. Recently, the coupling of different modeling strategies has also been explored. The flow-based model was coupled with GAN in [15] to obtain a likelihood for GAN; The VAE, flow-based model and GAN were coupled in [34] for more flexibility and efficiency. The main goal of deep generative models is to generate new data that are consistent with the underlying distribution of the available data. To achieve this, a specific

density model is not a necessity, e.g., GAN manages to focus on the mapping from a standard Gaussian to the desired data distribution without using the likelihood. Other than GANs, deep generative models usually provide a density model, e.g., the flow-based models actually define an invertible transport map between two random variables which yields an explicit push-forward measure. A common characteristic of deep generative models is that they employ neural networks to model the mapping between high-dimensional inputs and outputs whenever needed. Such a strategy is proved to be very effective for application problems although the models are usually not easy to analyze due to the strong nonlinearity induced by neural networks.

Classical density estimation techniques such as kernel density estimation and mixture of Gaussians, suffer severely from the curse of dimensionality, meaning that they are only effective for low-dimensional data. However, the approximation of high-dimensional distributions is often expected to alleviate the computational cost of sampling a complicated mathematical model. For example, a typical Uncertainty Quantification (UQ) model is a partial differential equation (PDE) subject to uncertainty. When studying rare events in such a system, we must have an effective strategy to reduce the number of samples since each sample corresponds to solving a PDE. One strategy is to use the reduced-order model to obtain the samples of the desired rare events followed by a density estimation step. The estimated distribution can then be coupled with the importance sampling technique for variance reduction [26, 12, 31]. Another important example is the variational Bayes [2]. Sampling strategies such as Markov Chain Monte Carlo (MCMC) become less effective as the number of dimensions increases. The variational Bayes approach, which seeks the optimal approximation of the distribution in a family of density models, may be more effective for high-dimensional problems than sampling strategies.

The available deep generative models usually focus on capturing the main features of the data instead of an accurate estimation of the density for that the dimensionality of the target data is often extremely high, e.g., high-resolution images that have millions of pixels. We are more interested in whether the strategies developed for deep generative models can be adapted as a density estimation technique with mathematical convergence. In [29], we coupled the real NVP [8] and the Knothe-Rosenblatt (KR) rearrangement to generate an invertible transport map, called KRnet, between the standard Gaussian and an arbitrary distribution. In numerical experiments, KRnet has demonstrated a much better algebraic convergence than the original real NVP with respect to the number of model parameters. The drawback of constructing a transport map is that the dimensionality needs to be kept unchanged, which limits KRnet to a relatively small number of dimensions. In this work, we intend to couple KRnet and variational autoencoder (VAE) to obtain a more general model called VAE-KRnet. The basic idea is to use KRnet to model the prior distribution of the latent random variable identified by VAE and generalize the underlying distribution of the encoder from Gaussian to an arbitrary one. We show that VAE-KRnet is more flexible than the canonical VAE by examining a linear model between the latent space and the data space. We then apply VAE-KRnet to the variational Bayes to approximate the posterior distribution. By varying the number of dimensions of the latent space from zero (KRnet) to d (VAE-KRnet), a wide range of data dimensions can be covered especially when

the problem admits a significant dimension reduction. One common problem in variational Bayes is the possible underestimation of variance because the minimization of the Kullback-Leibler divergence is more in favor of the first-order moments than the second-order moments, especially when the model capability is not strong enough. To alleviate this issue, we take into account the mutual information when searching the latent random variable. In the loss we balance the contribution of two terms: the maximization of the mutual information between the latent random variable and the original random variable, and the minimization of the KL divergence between the density model and the original distribution. The relative importance of these two terms will be adjusted by a weight parameter. By maximizing the mutual information we may keep more information from the region of low density and improve the estimation of the variance.

Our paper is organized as follows. We first present a brief description of KRnet in the next section. We discuss the coupling of VAE and KRnet in section 3, and apply VAE-KRnet to variational Bayes in section 4. In section 5, we study numerically the performance of VAE-KRnet, followed by a summary section.

2. KRNET - AN INVERTIBLE TRANSPORT MAP

Let $\mu_{\mathbf{Y}}$ and $\mu_{\mathbf{Z}}$ indicate the probability measures of random variables $\mathbf{Y}, \mathbf{Z} \in \mathbb{R}^n$ respectively. A transport map $T : \mathbf{Z} \rightarrow \mathbf{Y}$ is defined as $T_{\#}\mu_{\mathbf{Z}} = \mu_{\mathbf{Y}}$, where $T_{\#}\mu_{\mathbf{Z}}$ is the push-forward of the law $\mu_{\mathbf{Z}}$ of \mathbf{Z} such that $\mu_{\mathbf{Y}}(B) = \mu_{\mathbf{Z}}(T^{-1}(B))$ for every Borel set B [10]. The Knothe-Rosenblatt (K-R) rearrangement says that T may have a lower-triangular structure such that

$$\mathbf{z} = T^{-1}(\mathbf{y}) = f(\mathbf{y}) = \begin{bmatrix} f_1(y_1) \\ f_2(y_1, y_2) \\ \vdots \\ f_n(y_1, y_2, \dots, y_n) \end{bmatrix}, \quad (2.1)$$

which can be regarded as a limit of a sequence of optimal transport maps when the quadratic cost degenerates [5]. Combining the triangular structure of the K-R rearrangement and the technique real NVP [8], we have proposed an invertible mapping $\mathbf{z} = \hat{f}(\mathbf{y}) = \hat{T}^{-1}(\mathbf{y})$ such that $\hat{T}_{\#}\mu_{\mathbf{Z}}$ can be used as a model for density estimation when data are provided for \mathbf{Y} and a prior distribution is prescribed for \mathbf{Z} [28, 29]. The invertible transport map $\hat{f}(\cdot)$ is called KRnet. In reality, we may consider a block-triangular version of the K-R rearrangement for more flexibility. Consider a partition of $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_K)$, where $\mathbf{y}_i = (y_{i,1}, \dots, y_{i,m})$, where $1 \leq K \leq n$ and $1 \leq m \leq n$, and $\sum_{i=1}^K \dim(\mathbf{y}_i) = n$. The transport map $\hat{f}(\mathbf{y})$ then takes the following form:

$$\mathbf{z} = \hat{f}(\mathbf{y}) = \begin{bmatrix} \hat{f}_1(\mathbf{y}_1) \\ \hat{f}_2(\mathbf{y}_1, \mathbf{y}_2) \\ \vdots \\ \hat{f}_K(\mathbf{y}_1, \dots, \mathbf{y}_K) \end{bmatrix}. \quad (2.2)$$

Let $\mu_{\mathbf{Z}}(d\mathbf{z}) = p_{\mathbf{Z}}(\mathbf{z})d\mathbf{z}$, where $p_{\mathbf{Z}}(\mathbf{z})$ is the probability density function (PDF). The KRnet $\hat{f}(\cdot)$ induces the following density model

$$p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{Z}}(\hat{f}(\mathbf{y})) \left| \det \nabla_{\mathbf{y}} \hat{f}(\mathbf{y}) \right|, \quad (2.3)$$

which can be easily sampled as $\mathbf{Y} = \hat{f}^{-1}(\mathbf{Z})$, thanks to the invertibility of $\hat{f}(\cdot)$.

2.1. An overview of the layers in KRnet. The mathematical form of KRnet is an invertible composite mapping

$$\mathbf{z} = f(\mathbf{y}) = f_{[m]} \circ f_{[m-1]} \circ \dots \circ f_{[i]} \circ \dots \circ f_{[2]} \circ f_{[1]}(\mathbf{y}), \quad (2.4)$$

or

$$\mathbf{y} = f^{-1}(\mathbf{z}) = f_{[1]}^{-1} \circ f_{[2]}^{-1} \circ \dots \circ f_{[i]}^{-1} \circ \dots \circ f_{[m-1]}^{-1} \circ f_{[m]}^{-1}(\mathbf{z}), \quad (2.5)$$

where $f_{[i]}(\cdot)$ is a bijection that is often referred to as a layer. Simply speaking, KRnet modifies the data distribution of \mathbf{Y} step by step through a large number of intermediate simple bijections to make it eventually consistent with a prescribed distribution of \mathbf{Z} . We let $\mathbf{y}_{[0]} = \mathbf{y}$ indicate the initial state and $\mathbf{y}_{[i]} = f_{[i]} \circ \dots \circ f_{[1]}(\mathbf{y})$ an intermediate state. The main feature of KRnet is that the overall structure of the invertible mapping is lower (or upper) triangular. For the mapping from \mathbf{Y} to \mathbf{Z} , each dimension of \mathbf{Y} will be deactivated at a certain stage and remain fixed until all dimensions become inactive. On the other hand, the inverse mapping from \mathbf{Z} to \mathbf{Y} will activate the dimensions gradually. We now briefly introduce the layers used in KRnet, where each layer is a relatively simple bijection. More details about KRnet can be found in [29, 30].

Squeezing layer deactivates a certain number of components using a mask

$$\mathbf{q} = (\underbrace{1, \dots, 1}_k, \underbrace{0, \dots, 0}_{n-k}),$$

which means that the components $\mathbf{q} \odot \mathbf{y}_{[i]}$ will keep being updated and the rest components $(1 - \mathbf{q}) \odot \mathbf{y}_{[i]}$ will remain fixed from then on. Here \odot indicates the Hadamard product or component-wise product. So we deactivate the last $n - k$ components whenever needed.

Rotation layer provides a simple and trainable strategy to determine the dimensions that will be deactivated first. The rotation layer defines a rotation of the coordinate system through an orthonormal matrix for the current active dimensions:

$$\hat{\mathbf{y}}_{[i]} = \hat{\mathbf{W}}\mathbf{y}_{[i]} = \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{y}_{[i]} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{y}_{[i]},$$

where $\mathbf{W} \in \mathbb{R}^{k \times k}$ with k being the number of 1's in \mathbf{q} , and $\mathbf{I} \in \mathbb{R}^{(n-k) \times (n-k)}$ is an identity matrix, and $\mathbf{W} = \mathbf{L}\mathbf{U}$ is the LU decomposition of \mathbf{W} . The entries of \mathbf{L} and \mathbf{U} will be treated as trainable parameters except for the diagonal entries of \mathbf{L} which are equal to 1. Intuitively we expect the rotation may put the most important dimensions at the beginning, which need further modifications. One implementation issue is that we usually optimize the trainable entries of \mathbf{L} and \mathbf{U} directly for simplicity without enforcing the orthonormality of \mathbf{W} , which works well in practice.

Scale and bias layer provides a simplification of batch normalization which is defined as [17, 19]

$$\hat{\mathbf{y}}_{[i]} = \mathbf{a} \odot \mathbf{y}_{[i]} + \mathbf{b}, \quad (2.6)$$

where \mathbf{a} and \mathbf{b} are trainable, and initialized by the mean and standard deviation of data. After the initialization, \mathbf{a} and \mathbf{b} will be treated as regular trainable parameters that are independent of the data. The scale and bias layer helps to improve the conditioning of deep net.

Affine coupling layer is the most important layer for evolving the data. Consider a partition $\mathbf{y}_{[i]} = (\mathbf{y}_{[i],1}^\top, \mathbf{y}_{[i],2}^\top)^\top$ with $\mathbf{y}_{[i],1} \in \mathbb{R}^m$ and $\mathbf{y}_{[i],2} \in \mathbb{R}^{n-m}$. The affine coupling layer is defined as [29, 8]

$$\mathbf{z}_1 = \mathbf{y}_{[i],1}, \quad (2.7)$$

$$\mathbf{z}_2 = \mathbf{y}_{[i],2} \odot (1 + \alpha \tanh(\mathbf{s}(\mathbf{y}_{[i],1}))) + e^\beta \odot \tanh(\mathbf{t}(\mathbf{y}_{[i],1})), \quad (2.8)$$

where $\mathbf{s}, \mathbf{t} \in \mathbb{R}^{n-m}$ stand for scaling and translation functions depending only on $\mathbf{y}_{[i],1}$, $0 < \alpha < 1$ is a hyperparameter and $\beta \in \mathbb{R}^n$ is trainable. We modified the original affine coupling layer in real NVP [8] to improve the conditioning. Note that $\mathbf{y}_{[i],2}$ is updated linearly while the mappings $\mathbf{s}(\mathbf{y}_{[i],1})$ and $\mathbf{t}(\mathbf{y}_{[i],1})$ can be arbitrarily complicated, which are modeled as a neural network (NN),

$$(\mathbf{s}, \mathbf{t}) = \text{NN}(\mathbf{y}_{[i],1}). \quad (2.9)$$

Then the Jacobi matrix is lower-triangular, and an inverse can be easily computed. The two parts of $\mathbf{y}_{[i]}$ will be updated alternately by a sequence of affine coupling layers, e.g., at the next affine coupling layer, the first part will be modified while the second part remains fixed.

Nonlinear invertible layer defines a component-wise one-dimensional nonlinear mapping to map \mathbb{R} to itself. We decompose $\mathbb{R} = (-\infty, -a) \cup [-a, a] \cup (a, \infty)$ for $0 < a < \infty$. The intervals $(-\infty, -a) \cup (a, \infty)$ and $[-a, a]$ will be mapped to themselves. For $(-\infty, -a) \cup (a, \infty)$, a linear mapping is used, and for $[-a, a]$, a piecewise quadratic mapping is defined. More specifically, we define

$$z = \hat{F}(y) = \begin{cases} \beta(y - a) + a, & y \in (-\infty, -a) \\ \phi^{-1} \circ F \circ \phi(y), & y \in [-a, a] \\ \beta(y + a) - a, & y \in (a, \infty), \end{cases}$$

where $\phi : [-a, a] \rightarrow [0, 1]$ is an affine mapping, $\beta > 0$ is a scaling factor, and

$$F(x) = \int_0^x p(x) dx, \quad \forall x \in [0, 1], \quad (2.10)$$

where $p(x)$ can be regarded a PDF and $F(x)$ a cumulative distribution function. In particular, $p(x)$ will be defined as a piecewise linear function on a mesh of $[0, 1]$ such that $F(x)$ is a quadratic function whose inverse can be computed explicitly. Non-linear invertible layer provides a component-wise change of variable for a standard Gaussian prior distribution.

2.2. Main structure of KRnet. We are now ready to present the main structure of KRnet, which is illustrated in Figure 1. KRnet is mainly defined by two loops: outer loop $f_{[k]}^{\text{outer}}(\cdot)$ and inner loop $f_{[k,i]}^{\text{inner}}(\cdot)$ for a fixed k , where the outer loop has $K - 1$ stages, corresponding to the $K - 1$ mappings f_i in equation (2.2) with $i = 2, \dots, K$, and the inner loop has L stages, indicating the length of a chain of general coupling layers, see figure 1.

- Outer loop. Let $f_{[k]}^{\text{outer}}$ indicate one iteration of the outer loop. We have

$$\mathbf{z} = f(\mathbf{y}) = L_N \circ f_{[K-1]}^{\text{outer}} \circ \dots \circ f_{[1]}^{\text{outer}}(\mathbf{y}). \quad (2.11)$$

Let $\mathbf{y}_{[k]} = f_{[k]}^{\text{outer}}(\mathbf{y}_{[k-1]})$ with $\mathbf{y}_{[0]} = \mathbf{y}$, and $i = 1, \dots, K - 1$. Each $\mathbf{y}_{[k]} = (\mathbf{y}_{[k],1}, \dots, \mathbf{y}_{[k],K})$ has the same partition. The i th partition will remain unchanged after stage $K - i + 1$. For example, $\mathbf{y}_{[k],K}$ will be updated only when $k = 1$ and $\mathbf{y}_{[k],K-1}$ will be fixed when $k > 2$. This way, the number of

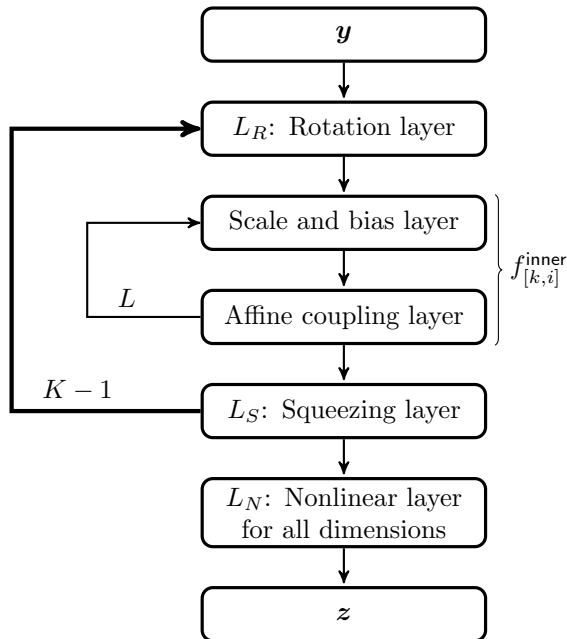


FIGURE 1. The flow chart of KRnet.

effective dimensions decreases as k increases. Once the outer loop is completed, the only active dimensions in $\mathbf{y}_{[K-1]} = (\mathbf{y}_{[K-1],1}, \dots, \mathbf{y}_{[K-1],K})$ will be $\mathbf{y}_{[K-1],1}$. We then apply the nonlinear invertible layer to all dimensions before the final output.

- Inner loop. The inner loop mainly consists of a sequence of general coupling layers $f_{[k,i]}^{\text{inner}}$, which includes one scale and bias layer and one affine coupling layer. We have

$$f_{[k]}^{\text{outer}} = L_S \circ f_{[k,L]}^{\text{inner}} \circ \dots \circ f_{[k,1]}^{\text{inner}} \circ L_R, \quad (2.12)$$

where L_R is a rotation layer, and L_S is a squeezing layer.

The main effectiveness of KRnet comes from the depth determined by both K and L . The rotation layers and the nonlinear invertible layers may be switched off initially to reduce the model complexity. Once the KRnet is trained, these layers can be switched on for further refinement.

3. COUPLING VAE AND KRNET

Due to the invertibility, KRnet maps a variable to another variable of the same dimension. To enhance its capability to deal with high-dimensional data, we will integrate it into the framework of variational autoencoder (VAE), which provides a dimension reduction technique for density estimation. We employ VAE to identify the latent space and KRnet to generalize the modeling for the prior distribution and the encoder.

3.1. Variational autoencoder (VAE). We briefly recall the variational autoencoder [18]. Assume that there exists a latent random variable $\mathbf{X} \in \mathbb{R}^d$ with $d \ll n$

with a marginal distribution $p_{\mathbf{X},\theta}(\mathbf{x})$, where θ includes the model parameters. The joint distribution $p_{\mathbf{X},\mathbf{Y},\theta}$ of \mathbf{X} and \mathbf{Y} is then described by the conditional distribution $p_{\mathbf{Y}|\mathbf{X},\theta}(\mathbf{y}|\mathbf{x})$, i.e., $p_{\mathbf{X},\mathbf{Y},\theta} = p_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}$.

The target is to approximate the posterior distribution $p_{\mathbf{X}|\mathbf{Y}^{(i)},\theta}(\mathbf{x}|\mathbf{y}^{(i)})$ which will be modeled by a family of parameterized PDFs $q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi}(\mathbf{x}|\mathbf{y}^{(i)})$. Here we add a superscript $^{(i)}$ to emphasize that the random variable $\mathbf{Y}^{(i)}$ corresponds to one sample in the training set. The optimal parameters θ and ϕ are determined by minimizing the KL divergence

$$\begin{aligned} & D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \| p_{\mathbf{X}|\mathbf{Y}^{(i)},\theta}) \\ &= D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \| p_{\mathbf{X},\theta}) - \int q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \log p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta} d\mathbf{x} + \log p_{\mathbf{Y}^{(i)}} \geq 0. \end{aligned} \quad (3.1)$$

The minimization of $D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \| p_{\mathbf{X}|\mathbf{Y}^{(i)},\theta})$ is equivalent to the maximization of the variational lower bound of $\log p_{\mathbf{Y}^{(i)}}$, which is defined as

$$\mathcal{L}_{\theta,\phi}(\mathbf{y}^{(i)}) = -D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \| p_{\mathbf{X},\theta}) + \int q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} \log p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta} d\mathbf{x}. \quad (3.2)$$

If there are N samples in the training set, the variational lower bound of the log-likelihood $\log p_{\mathbf{Y}}$ is

$$\mathcal{L}_{\theta,\phi}(\mathbf{y}) = \sum_{i=1}^N \mathcal{L}_{\theta,\phi}(\mathbf{y}^{(i)}), \quad (3.3)$$

where \mathbf{y} includes all the data $\{\mathbf{y}^{(i)}\}_{i=1}^N$ in the training set.

To apply VAE we need to specify three PDF models respectively for $p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta}$, $q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi}$ and $p_{\mathbf{X},\theta}(\mathbf{x})$. In the canonical VAE, the following models are used:

$$p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta} = \mathcal{N}(\boldsymbol{\mu}_{\text{de},\theta}(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}_{\text{de},\theta}^{\odot 2}(\mathbf{x}))), \quad (3.4)$$

$$q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi} = \mathcal{N}(\boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}^{(i)}), \text{diag}(\boldsymbol{\sigma}_{\text{en},\phi}^{\odot 2}(\mathbf{y}^{(i)}))), \quad (3.5)$$

$$p_{\mathbf{X},\theta} = \mathcal{N}(0, \mathbf{I}), \quad (3.6)$$

where $^{\odot 2}$ means that the square operation is component-wise. From the viewpoint of dimension reduction, it is often a good choice to assume that $p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta}$ is Gaussian with independent components. The posterior distribution $p_{\mathbf{X}|\mathbf{Y}^{(i)},\theta}$ is in general intractable, and an approximation model $q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi}$ is needed, which is also chosen as a multivariate Gaussian with independent components. Then $(\boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}), \boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y}))$ serves as the encoder and $(\boldsymbol{\mu}_{\text{de},\theta}(\mathbf{x}), \boldsymbol{\sigma}_{\text{de},\theta}(\mathbf{x}))$ serves as the decoder. The prior distribution $p_{\mathbf{X},\theta}$ is simply assumed to be a standard Gaussian $\mathcal{N}(0, \mathbf{I})$. Furthermore, both encoder and decoder are modeled by neural networks.

3.2. VAE for a linear model. VAE has three components: the prior, the encoder and the decoder, each of which is either a simple Gaussian or a diagonal Gaussian. Let us first examine the relation of these three components in terms of a linear model for dimension reduction

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \boldsymbol{\xi}, \quad (3.7)$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$, $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, $\mathbf{A} \in \mathbb{R}^{n \times d}$, and $\boldsymbol{\xi}$ is independent of \mathbf{X} . We regard \mathbf{X} as a latent random variable with $d \ll n$. The joint distribution of \mathbf{X} and \mathbf{Y} is

$$p_{\mathbf{X},\mathbf{Y}} = p_{\mathbf{X}} \cdot \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2 \mathbf{I}) \quad (3.8)$$

After the decoder $p_{\mathbf{Y}|\mathbf{X}} = \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2\mathbf{I})$ is specified, we look at the relation between $p_{\mathbf{X}}$ and $q_{\mathbf{Y}|\mathbf{X}}$:

Lemma 3.1. *Let $p_{\mathbf{X}} = \mathcal{N}(0, \Sigma_{\mathbf{X}})$ in model (3.7) with $\Sigma_{\mathbf{X}}$ being positive definite. From the modeling point of view, the encoder $q_{\mathbf{X}|\mathbf{Y}} = \mathcal{N}(\boldsymbol{\mu}_{\text{en}}(\mathbf{y}), \boldsymbol{\sigma}_{\text{en}}^{\odot 2}(\mathbf{y}))$ of the canonical VAE is able to exactly recover the true posterior distribution $p_{\mathbf{X}|\mathbf{Y}}$.*

Proof. The prior is chosen as $p_{\mathbf{Z}} = \mathcal{N}(0, \mathbf{I})$ in VAE, implying that we may consider a transformed model

$$\mathbf{Y} = \mathbf{A}(\mathbf{B}\mathbf{U}\mathbf{Z}) + \boldsymbol{\xi},$$

where $\mathbf{X} = \mathbf{B}\mathbf{U}\mathbf{Z}$ with $\mathbf{B}, \mathbf{U} \in \mathbb{R}^{d \times d}$ and \mathbf{U} being a unitary matrix. The reason that we include the unitary matrix \mathbf{U} will be clear later. It is easy to see that the variable $\mathbf{Z}|\mathbf{y}$ is Gaussian subject to a covariance matrix

$$\Sigma_{\mathbf{Z}|\mathbf{y}} = (\mathbf{I} + \sigma^{-2}\mathbf{U}^{\top}\mathbf{B}^{\top}\mathbf{A}^{\top}\mathbf{A}\mathbf{B}\mathbf{U})^{-1}. \quad (3.9)$$

If the encoder of VAE is able to exactly recover $\Sigma_{\mathbf{Z}|\mathbf{y}}$, we need the existence of a linear mapping $\mathbf{B}\mathbf{U}$ such that $\Sigma_{\mathbf{Z}|\mathbf{y}}$ is diagonal. Note that

$$\text{Cov}(\mathbf{X}) = \Sigma_{\mathbf{X}} = \mathbf{B}\mathbf{U}\text{Cov}(\mathbf{Z})\mathbf{U}^{\top}\mathbf{B}^{\top} = \mathbf{B}\mathbf{B}^{\top},$$

which means that we can let $\mathbf{B} = \Sigma_{\mathbf{X}}^{1/2}$. According to the spectral theorem of symmetric matrices, we know there exists a unitary matrix \mathbf{U} such that $\mathbf{U}^{\top}\mathbf{B}^{\top}\mathbf{A}^{\top}\mathbf{A}\mathbf{B}\mathbf{U}$ is diagonal. This concludes the lemma. \square \square

Remark 3.2. *For the linear model (3.7) with a Gaussian prior, the canonical VAE is able to model the posterior, where the encoder needs to take care of three mappings: (1) the mapping \mathbf{B} , and (2) the rotation \mathbf{U} , and (3) the matrix operations in equation (3.9). The mapping \mathbf{B} transfers a general Gaussian to a standard one, and the rotation \mathbf{U} makes the covariance matrix of $\mathbf{Z}|\mathbf{y}$ diagonal. Under the assumption that the diagonal decoder provides a reasonable model for dimension reduction, we know from this case study that the encoder needs to balance the following two issues:*

- (1) Map the prior distribution $p_{\mathbf{X}}$ to a standard Gaussian $\mathcal{N}(0, \mathbf{I})$;
- (2) Map the posterior distribution $p_{\mathbf{X}|\mathbf{Y}}$ to a diagonal Gaussian.

So the effectiveness of the canonical VAE depends on how well these two issues can coexist, which is obviously problem dependent.

3.3. Generalize the prior. Assume that $\mathbf{X} \sim p_G = \mathcal{N}(0, \mathbf{I})$ following the canonical VAE. We introduce another random variable \mathbf{Z} satisfying $\mathbf{X} = f_{\text{pr},\beta}(\mathbf{Z})$, where $f_{\text{pr},\beta}(\cdot)$ is a nonlinear bijection with β being the model parameter, e.g., KRnet. We have $p_{\mathbf{Z},\beta}(\mathbf{z}) = p_G(f_{\text{pr},\beta}(\mathbf{z}))|\nabla_{\mathbf{z}}f_{\text{pr},\beta}(\mathbf{z})|$. We now compare the two cases, where the latent spaces are defined by \mathbf{X} and \mathbf{Z} respectively. We also assume that $q_{\mathbf{X}|\mathbf{Y},\phi}$ and $q_{\mathbf{Z}|\mathbf{Y},\phi}$ are defined by the same model, i.e., Gaussian, and so are $p_{\mathbf{Y}|\mathbf{X},\theta}$ and $p_{\mathbf{Y}|\mathbf{Z},\theta}$. In other words, only the model for the prior is changed.

Lemma 3.3. *If $f_{\text{pr},\beta}(\cdot)$ induces a density model that is able to approximate any d -dimensional PDF arbitrarily well, then there exists a parameter $\beta = \tilde{\beta}$ such that VAE can reach a larger variational lower bound in terms of the random variable \mathbf{Z} rather than \mathbf{X} .*

Proof. Noting that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \mathcal{L}_{\phi,\theta}^{\mathbf{X}}(\mathbf{y}) = \lim_{N \rightarrow \infty} \sum_{i=1}^N \frac{1}{N} \mathcal{L}_{\phi,\theta}^{\mathbf{X}}(\mathbf{y}^{(i)}) = \mathbb{E}_{p_{\mathbf{Y}}} \mathcal{L}_{\phi,\theta}^{\mathbf{X}}(\mathbf{Y}), \quad (3.10)$$

where the superscript \mathbf{X} indicates that the latent space is defined by $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I})$ and

$$\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi, \theta}^{\mathbf{X}}(\mathbf{Y})] = -D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}, \phi} p_{\mathbf{Y}} \| p_{\mathbf{X}, \theta} p_{\mathbf{Y}}) + \mathbb{E}_{p_{\mathbf{Y}} q_{\mathbf{X}|\mathbf{Y}, \phi}} [\log p_{\mathbf{Y}|\mathbf{X}, \theta}]. \quad (3.11)$$

For simplicity, we consider $\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi^*, \theta^*}^{\mathbf{X}}(\mathbf{Y})]$ instead of the average of $\mathcal{L}_{\phi, \theta}^{\mathbf{X}}(\mathbf{y}^{(i)})$, where

$$(\phi^*, \theta^*) = \arg \max_{\phi, \theta} \mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi, \theta}^{\mathbf{X}}(\mathbf{Y})]. \quad (3.12)$$

By definition, we have

$$\begin{aligned} & \mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi^*, \theta^*}^{\mathbf{X}}(\mathbf{Y})] - \mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi^*, \theta^*, \beta}^{\mathbf{Z}}(\mathbf{Y})] \\ &= D_{\text{KL}}(q_{\mathbf{Z}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \| p_{\mathbf{Z}, \beta} p_{\mathbf{Y}}) - D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \| p_G p_{\mathbf{Y}}), \end{aligned}$$

since only the prior depends on β and the encoders and decoders are the same for both \mathbf{X} and \mathbf{Z} . We have

$$\begin{aligned} & D_{\text{KL}}(q_{\mathbf{Z}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \| p_{\mathbf{Z}, \beta} p_{\mathbf{Y}}) - D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \| p_G p_{\mathbf{Y}}) \\ &= - \int q_{\mathbf{Z}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \log p_{\mathbf{Z}, \beta} dz d\mathbf{y} + \int q_{\mathbf{X}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}} \log p_G dx d\mathbf{y} \\ &= - \int q_{\mathbf{Z}, \phi^*} \log p_{\mathbf{Z}, \beta} dz + \int q_{\mathbf{X}, \phi^*} \log p_G dx \\ &= D_{\text{KL}}(q_{\mathbf{Z}, \phi^*} \| p_{\mathbf{Z}, \beta}) - D_{\text{KL}}(q_{\mathbf{X}, \phi^*} \| p_G), \end{aligned}$$

where $q_{\mathbf{Z}, \phi^*}$ and $q_{\mathbf{X}, \phi^*}$ refer to the same marginal distribution because by assumption the joint distributions $q_{\mathbf{Z}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}}$ and $q_{\mathbf{X}|\mathbf{Y}, \phi^*} p_{\mathbf{Y}}$ are the same. So if $q_{\mathbf{X}, \phi^*} \neq p_G$, there always exists $\tilde{\beta}$ such that

$$D_{\text{KL}}(q_{\mathbf{Z}, \phi^*} \| p_{\mathbf{Z}, \tilde{\beta}}) < D_{\text{KL}}(q_{\mathbf{X}, \phi^*} \| p_G)$$

since by assumption the density model induced by $\mathbf{X} = f_{\text{pr}, \beta}(\mathbf{Z})$ can approximate the PDF $q_{\mathbf{Z}, \phi^*}$ arbitrarily well. We then have

$$\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi^*, \theta^*, \tilde{\beta}}^{\mathbf{Z}}(\mathbf{Y})] > \mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\phi^*, \theta^*}^{\mathbf{X}}(\mathbf{Y})],$$

implying that \mathbf{Z} provides a better latent space than \mathbf{X} . \square \square

3.4. Generalize the encoder. We now look at the encoder $q_{\mathbf{X}|\mathbf{Y}}$. Note

$$p_{\mathbf{Y}} p_{\mathbf{X}|\mathbf{Y}} = p_{\mathbf{X}, \mathbf{Y}} = p_{\mathbf{X}} p_{\mathbf{Y}|\mathbf{X}}, \quad (3.13)$$

where $p_{\mathbf{X}, \mathbf{Y}}$ is the joint distribution. In this equation, three PDFs, i.e., $p_{\mathbf{Y}|\mathbf{X}}$, $p_{\mathbf{X}|\mathbf{Y}}$ and $p_{\mathbf{X}}$, will be modeled or approximated by Gaussians in the canonical VAE as shown in equations (3.4)-(3.6). Although it is straightforward to assume that $p_{\mathbf{Y}|\mathbf{X}}$ is a diagonal Gaussian from the viewpoint of model reduction, modeling $p_{\mathbf{X}|\mathbf{Y}}$ as a diagonal Gaussian is mainly for tractability. Let us consider the following optimization problem

$$(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) = \min_{(\boldsymbol{\mu}, \boldsymbol{\Sigma})} D_{\text{KL}}[p_{\mathbf{X}|\mathbf{Y}} \| \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})], \quad (3.14)$$

which yields the optimal Gaussian that approximates $p_{\mathbf{X}|\mathbf{Y}}$. More specifically, we have

$$\boldsymbol{\mu}^* = \int p_{\mathbf{X}|\mathbf{Y}} \mathbf{x} d\mathbf{x} = \mathbb{E}_{p_{\mathbf{X}|\mathbf{Y}}}[\mathbf{X}]. \quad (3.15)$$

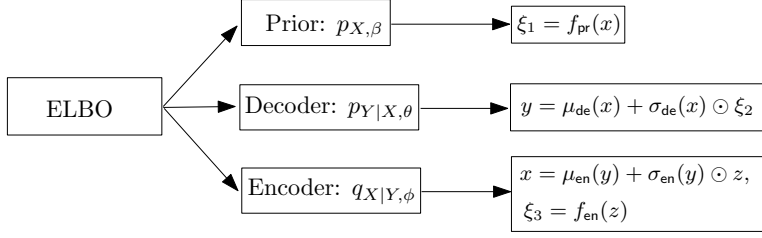


FIGURE 2. Generalized variational autoencoder, where two flow-based generative models $f_{\text{pr}}(\cdot)$ and $f_{\text{en}}(\cdot)$ are integrated.

and

$$\Sigma^* = \int p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top d\mathbf{x} = \mathbb{E}_{p_{\mathbf{X}|\mathbf{Y}}} [(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top]. \quad (3.16)$$

Equation (3.16) shows that to approximate $p_{\mathbf{X}|\mathbf{Y}}$ with a Gaussian, the encoder (3.5) with a diagonal covariance matrix is in general not enough.

A straightforward way to generalize the encoder of the canonical VAE is to consider a full covariance matrix, i.e., $\mathcal{N}(\boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}), \boldsymbol{\Sigma}_{\text{en},\phi}(\mathbf{y}))$. We here propose a simpler strategy. We let

$$f_{\text{en},\alpha} \left(\frac{\mathbf{X}|\mathbf{y} - \boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y})}{\boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y})} \right) \sim \mathcal{N}(0, \mathbf{I}), \quad (3.17)$$

where $f_{\text{en},\alpha}$ is a KRnet. In other words, we can write

$$\mathbf{X}|\mathbf{y} = \boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}) + \boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y}) \odot f_{\text{en},\alpha}^{-1}(\boldsymbol{\xi}), \quad (3.18)$$

where $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})$. If we let $f_{\text{en},\alpha}(\cdot)$ be an identity mapping, the original encoder (3.5) is recovered. Let us look at the linear model (3.7) again. If the prior $p_{\mathbf{X}} = \mathcal{N}(\boldsymbol{\mu}_{\text{pr}}, \boldsymbol{\Sigma}_{\text{pr}})$ is an arbitrary Gaussian, it can be obtained that the covariance matrix for $\mathbf{X}|\mathbf{y}$ is

$$\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_{\text{pr}}^{-1} + \sigma^{-2} \mathbf{A}^\top \mathbf{A})^{-1}.$$

Let $\boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}) = \mathbb{E}_{p_{\mathbf{X}|\mathbf{Y}}}[\mathbf{X}]$, $\boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y}) = \mathbf{1}$ and $f_{\text{en},\alpha}^{-1}(\boldsymbol{\xi}) = \boldsymbol{\Sigma}^{1/2} \boldsymbol{\xi}$. The encoder (3.18) is able to recover $p_{\mathbf{X}|\mathbf{Y}}$ exactly for the linear model (3.7) with any Gaussian priors, where the prior can be modeled directly by $f_{\text{pr},\beta}(\cdot)$ and the correlation of $\mathbf{X}|\mathbf{y}$ can be taken care of by $f_{\text{en},\alpha}^{-1}(\cdot)$, in contrast to the canonical VAE, where the diagonal Gaussian encoder needs to achieve what both $f_{\text{pr},\beta}(\cdot)$ and $f_{\text{en},\alpha}^{-1}(\cdot)$ do. In this sense, the model becomes more flexible since both $f_{\text{pr},\beta}(\cdot)$ and $f_{\text{en},\alpha}^{-1}(\cdot)$ can help to maintain the diagonal form of the encoder.

3.5. VAE-KRnet. To this end, we have a simple strategy to couple VAE and KRnet. Within the framework of VAE, we keep the original decoder $\mathcal{N}(\boldsymbol{\mu}_{\text{de},\theta}(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}_{\text{de},\theta}^{\odot 2}(\mathbf{x})))$, but generalize the original prior p_G and the encoder $\mathcal{N}(\boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}), \text{diag}(\boldsymbol{\sigma}_{\text{en},\phi}^{\odot 2}(\mathbf{y})))$ by incorporating two KRnets $f_{\text{pr},\beta}(\cdot)$ and $f_{\text{en},\alpha}(\cdot)$, respectively, as demonstrated in figure 2.

We now define the variational lower bound of VAE-KRnet using the following PDFs:

$$p_{\mathbf{X},\beta} = p_G(f_{\text{pr},\beta}(\mathbf{x}))|\nabla_{\mathbf{x}}f_{\text{pr},\beta}(\mathbf{x})|, \quad (3.19)$$

$$q_{\mathbf{X}|\mathbf{Y},\phi,\alpha} = p_G\left(f_{\text{en},\alpha}\left(\frac{\mathbf{x} - \boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y})}{\boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y})}\right)\right)|\nabla_{\mathbf{x}}f_{\text{en},\alpha}(\mathbf{x})|, \quad (3.20)$$

$$p_{\mathbf{Y}|\mathbf{X},\theta} = \mathcal{N}(\boldsymbol{\mu}_{\text{de},\theta}(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}_{\text{de},\theta}^{\odot 2}(\mathbf{x}))). \quad (3.21)$$

The variational lower bound can be easily approximated by the Monte Carlo method. In particular, the so-called reparameterization trick [7] can be employed. Using equation (3.18), the samples for the PDF $q_{\mathbf{X}|\mathbf{Y},\phi,\alpha}(\mathbf{x}|\mathbf{y}^{(i)})$ can be represented as

$$\mathbf{x}^{(i,j)} = \boldsymbol{\mu}_{\text{en},\phi}(\mathbf{y}^{(i)}) + \boldsymbol{\sigma}_{\text{en},\phi}(\mathbf{y}^{(i)}) \odot \mathbf{z}^{(j)}, \quad j = 1, \dots, J, \quad (3.22)$$

where $\mathbf{z}^{(j)} = f_{\text{en},\alpha}^{-1}(\boldsymbol{\xi}^{(j)})$ and $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})$. The variational lower bound (3.2) will be approximated as

$$\mathcal{L}_{\theta,\phi,\beta,\alpha}^{\mathbf{X}}(\mathbf{y}^{(i)}) \approx \frac{1}{J} \sum_{j=1}^J \log \frac{p_{\mathbf{Y}^{(i)}|\mathbf{X},\theta}(\mathbf{y}^{(i)}|\mathbf{x}^{(i,j)})p_{\mathbf{X},\beta}(\mathbf{x}^{(i,j)})}{q_{\mathbf{X}|\mathbf{Y}^{(i)},\phi,\alpha}(\mathbf{x}^{(i,j)}|\mathbf{y}^{(i)})}. \quad (3.23)$$

For simplicity, we can just let $J = 1$ by noting that

$$\begin{aligned} \mathbb{E}_{p_{\mathbf{Y}}}[\mathcal{L}^{\mathbf{X}}] &= \mathbb{E}_{p_{\mathbf{Y}}q_{\mathbf{X}|\mathbf{Y}}} \left[\log \frac{p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})p_{\mathbf{X}}}{q_{\mathbf{X}|\mathbf{Y}}} \right] \\ &\approx \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\theta,\phi,\beta,\alpha}^{\mathbf{X}}(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) = \frac{1}{N} \mathcal{L}_{\theta,\phi,\beta,\alpha}^{\mathbf{X}}(\mathbf{y}), \end{aligned} \quad (3.24)$$

where $(\mathbf{y}^{(i)}, \mathbf{x}^{(i)})$ corresponds to one sample from the joint PDF $p_{\mathbf{Y}}q_{\mathbf{X}|\mathbf{Y}}$.

Remark 3.4. *It is seen that VAE-KRnet naturally integrates KRnet into the framework of VAE. Compared to other flow-based models, KRnet has its advantages. We here mainly comment on the invertibility of the transport map. Most flow-based generative models, which are able to provide an explicit likelihood, can be regarded as a transport map which requires either discrete or continuous invertibility. Discrete invertibility includes NICE [7], real NVP [8], planar flow [25], inverse autoregressive flow [21], Sylvester flow [3], KRnet [29], etc. Continuous invertibility mainly refers to neural ODE [6] and its variants subject to either augmentation [9] or regularization [32, 11]. The continuous invertibility can only be achieved in terms of the original ODE instead of the discretized ODE, which implies that the gradient of the loss may not be accurately computed through the adjoint method. Discrete invertibility such as the planar flow and the Sylvester flow depends on the inverse of an activation function, e.g., tanh. The mapping from a finite interval to an infinite interval may have an issue on the robustness. To enhance VAE with flow-based generative models, we expect that the invertibility of the flow-based generative models can be easily maintained such that the overall model is relatively robust. The inverse autoregressive model and the real NVP, which is a generalization of NICE, can achieve discrete invertibility easily and exactly. However, the inverse autoregressive flow is more like a probabilistic model instead of a dynamical one, where the model structure is determined by the decomposition of the PDF in terms of conditional distributions. The real NVP defines a transport map, but its drawback is the information exchange among dimensions becomes less effective as the*

depth increases. By integrating the triangular structure of the Knothe-Rosenblatt rearrangement, KRnet, as a generalization of the real NVP, alleviates this issue such that the model capability has been significantly improved while the exact invertibility is kept.

4. DENSITY APPROXIMATION

VAE-KRnet provides a family of probability density models that may be used for density estimation or approximation when data are available or the PDF is known up to a constant. If dimension reduction is not considered, we use KRnet instead of VAE-KRnet. The configuration of KRnet is a trade-off between the number K of outer-loop iterations and the number L of inner-loop iterations. For a large n , we need to reduce at least one of the two numbers K and L for KRnet such that the model is affordable. Another way to deal with high dimensionality is to consider VAE-KRnet, where a low-dimensional latent space is sought. Since in reality problem solutions often admits a low-dimensional approximation, VAE-KRnet provides a mechanism to adapt the properties of the problem into density estimation. For simplicity, we use the subscript $*_{\theta}$ to indicate a PDF model with a general model parameter θ in this section.

4.1. Metrics for seeking the latent random variable. The canonical VAE seeks the latent random variable based on available data by minimizing the KL divergence between $q_{\mathbf{X}|\mathbf{Y}}$ and $p_{\mathbf{X}|\mathbf{Y}}$ for any \mathbf{y} (see equation (3.1)). Another way to achieve this is as follows:

Lemma 4.1. *Maximizing $\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}^{\mathbf{X}}]$ in VAE is equivalent to minimizing the following K-L divergence:*

$$D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}\|p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}), \quad (4.1)$$

which yields $q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}} = p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}$ when the minimum is reached.

Proof. Equation (4.1) can be rewritten as:

$$\begin{aligned} & D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}\|p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}) \\ &= \int q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}} \log \frac{q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}}{p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}} d\mathbf{x}d\mathbf{y} \\ &= \int q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}} \left(\log \frac{q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}}{p_{\mathbf{X}}p_{\mathbf{Y}}} + \log \frac{p_{\mathbf{Y}}}{p_{\mathbf{Y}|\mathbf{X}}} \right) d\mathbf{x}d\mathbf{y} \\ &= -\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}^{\mathbf{X}}] + \int q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}} \log p_{\mathbf{Y}} d\mathbf{x}d\mathbf{y} \\ &= -\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}^{\mathbf{X}}] - h(\mathbf{Y}) \geq 0. \end{aligned}$$

So minimizing $D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}\|p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}})$ is the same as maximizing $\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}^{\mathbf{X}}]$, since the differential entropy $h(\mathbf{Y})$ is not related to the models. \square \square

When we only have the samples of $p_{\mathbf{Y}}$, it is convenient to maximize the variational lower bound $\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}^{\mathbf{X}}]$. We also note that the relation $q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}} = p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}$ can be obtained by minimizing

$$D_{\text{KL}}(p_{\mathbf{Y}|\mathbf{X}}p_{\mathbf{X}}\|q_{\mathbf{X}|\mathbf{Y}}p_{\mathbf{Y}}), \quad (4.2)$$

where we switch the two distributions in equation (4.1). Due to the asymmetry of the K-L divergence, equations (4.1) and (4.2) are different from the computation

point of view although they yield the same minimum and minimizer. Equation (4.2) is useful when $p_{\mathbf{Y}}$ is known up to a constant and data are not available. For this case the K-L divergence (4.2) may be approximated by the samples generated in terms of $p_{\mathbf{X}}$ and $p_{\mathbf{Y}|\mathbf{X}}$.

4.2. When data are available. For this case, we maximize the variational lower bound using the data of \mathbf{Y} . After obtaining the trained PDFs $p_{\mathbf{X},\theta}$ and $p_{\mathbf{Y}|\mathbf{X},\theta}$, we can approximate the marginal PDF of \mathbf{Y} :

$$p_{\mathbf{Y},\theta}(\mathbf{y}) = \mathbb{E}_{p_{\mathbf{X},\theta}}[p_{\mathbf{Y}|\mathbf{X},\theta}] \approx \frac{1}{N} \sum_{i=1}^N p_{\mathbf{Y}|\mathbf{X},\theta}(\mathbf{y}|\mathbf{x}^{(i)}), \quad (4.3)$$

where the samples $\mathbf{x}^{(i)} = f_{\text{pr},\beta}^{-1}(\boldsymbol{\xi}^{(i)})$ with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A more efficient way to compute $p_{\mathbf{Y},\theta}(\mathbf{y})$ is

$$p_{\mathbf{Y},\theta}(\mathbf{y}) = \mathbb{E}_{q_{\mathbf{X}|\mathbf{Y},\theta}} \left[\frac{p_{\mathbf{Y}|\mathbf{X},\theta} p_{\mathbf{X},\theta}}{q_{\mathbf{X}|\mathbf{Y},\theta}} \right] \approx \frac{1}{N} \sum_{i=1}^N \frac{p_{\mathbf{Y}|\mathbf{X},\theta}(\mathbf{y}|\mathbf{x}^{(i)}) p_{\mathbf{X},\theta}(\mathbf{x}^{(i)})}{q_{\mathbf{X}|\mathbf{Y},\theta}(\mathbf{x}^{(i)}|\mathbf{y})} \quad (4.4)$$

which uses the posterior to implement importance sampling. If the posterior is well approximated, the variance should be small such that less samples are needed compared to equation (4.3).

4.3. When PDF is available. For many cases, we need to sample or approximate an arbitrary PDF known up to a constant, e.g., the posterior in Bayesian inference. As an alternative of sampling methods such as MCMC, variational Bayes methods seek the best approximation of the posterior within a given family of density models. One classical variational Bayes approach is the mean-field approximation, where mutual independence is assumed for a partition of the latent random variable. In this work, we use VAE-KRnet as the variational distribution. We pay particular attention to one common problem of variational Bayes methods, which is the underestimation of variance. To alleviate this issue, we introduce the maximization of mutual information when seeking the latent random variable of VAE.

4.3.1. KRnet. Let $p_{\mathbf{Y}} = C^{-1} \hat{p}_{\mathbf{Y}}$ be a PDF, where C is an unknown normalization constant, i.e., $\int \hat{p}_{\mathbf{Y}} d\mathbf{y} = C$. Let $q_{\mathbf{Y},\theta}$ be a PDF model given by KRnet. To train the KRnet, we consider the KL divergence between $q_{\mathbf{Y},\theta}$ and $p_{\mathbf{Y}}$, where the unknown constant C will shown up as a shift that does not affect the minimization, i.e.,

$$D_{\text{KL}}(q_{\mathbf{Y},\theta} \| p_{\mathbf{Y}}) = \int q_{\mathbf{Y},\theta} \log \frac{q_{\mathbf{Y},\theta}}{\hat{p}_{\mathbf{Y}}} d\mathbf{y} + \log C = \mathcal{D}_{\theta}^{\text{pdf}}(\hat{p}_{\mathbf{Y}}) + \log C. \quad (4.5)$$

Minimizing $D_{\text{KL}}(q_{\mathbf{Y},\theta} \| p_{\mathbf{Y}})$ is equivalent to minimizing $\mathcal{D}_{\theta}^{\text{pdf}}$. In general, $\mathcal{D}_{\theta}^{\text{pdf}}$ needs to be approximated by sampling, which is trivial thanks to the generative modeling. Noting that KRnet corresponds to an invertible mapping $\mathbf{Z} = f(\mathbf{Y})$ such that $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we can easily apply the reparameterization trick, i.e.,

$$\mathcal{D}_{\theta}^{\text{pdf}}(\hat{p}_{\mathbf{Y}}) = \int q_{\mathbf{Z}} \log \frac{q_{\mathbf{Y},\theta}(f^{-1}(\mathbf{z}))}{\hat{p}_{\mathbf{Y}}(f^{-1}(\mathbf{z}))} dz \approx \frac{1}{N} \sum_{i=1}^N \log \frac{q_{\mathbf{Y},\theta}(f^{-1}(\mathbf{z}^{(i)}))}{\hat{p}_{\mathbf{Y}}(f^{-1}(\mathbf{z}^{(i)}))}. \quad (4.6)$$

So the training set is simply a set $\{\mathbf{z}^{(i)}\}_{i=1}^N$ of samples from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

4.3.2. *VAE-KRnet*. If dimension reduction is considered, the latent random variable \mathbf{X} is taken into account such that the PDF model is given by VAE-KRnet. For this case, let us specify $p_{\mathbf{Y}}$ and $p_{\mathbf{X}|\mathbf{Y}}$ and use $q_{\mathbf{Y}|\mathbf{X}}$ to indicate the approximation of $p_{\mathbf{Y}|\mathbf{X}}$. We then minimize the following objective function

$$-D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X},\theta}p_{\mathbf{Y}}) + \lambda D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X}|\mathbf{Y},\theta}p_{\mathbf{Y}}), \quad (4.7)$$

where the first term corresponds to the mutual information between \mathbf{X} and \mathbf{Y} , and λ is the Lagrange multiplier of the constraint $D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X}|\mathbf{Y},\theta}p_{\mathbf{Y}}) = 0$. The second term here acts as a regularization term with $\lambda > 0$. Minimizing the given objective function will maximize the mutual information between \mathbf{X} and \mathbf{Y} subject to the constraint that $D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X}|\mathbf{Y},\theta}p_{\mathbf{Y}})$ is small as much as possible. Removing the normalization constant in $p_{\mathbf{Y}}$ as in the previous section, we define

$$\begin{aligned} & \mathcal{E}_{\theta}^{\text{pdf}}(\hat{p}_{\mathbf{Y}}) \\ &= \int q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta} \left[\log \left(\frac{q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}}{\hat{p}_{\mathbf{Y}}} \right)^{\lambda-1} + \log p_{\mathbf{X},\theta} - \log p_{\mathbf{X}|\mathbf{Y},\theta}^{\lambda} \right] dx dy, \quad (4.8) \end{aligned}$$

where $\lambda > 1$. When we decrease λ from ∞ , the term $\log \left(\frac{q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}}{\hat{p}_{\mathbf{Y}}} \right)^{\lambda-1} = 0$ at $\lambda = 1$. Then $\hat{p}_{\mathbf{Y}}$ disappears from the loss function and the problem becomes ill-posed, meaning that the minimum will be $-\infty$. When $\lambda < 1$, the regularization term is even weaker, and the problem will be still ill-posed. Similar to equation (4.6), we can approximate $\mathcal{E}_{\theta}^{\text{pdf}}(\hat{p}_{\mathbf{Y}})$ using the reparameterization trick.

When λ goes to infinity, minimizing the objective equation (4.7) is equivalent to minimize directly the KL divergence $D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X}|\mathbf{Y},\theta}p_{\mathbf{Y}})$. It is seen $p_{\mathbf{Y}} = \mathbb{E}_{p_{\mathbf{X},\theta}}[q_{\mathbf{Y}|\mathbf{X},\theta}]$ as long as $D_{\text{KL}}(q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}\|p_{\mathbf{X}|\mathbf{Y},\theta}p_{\mathbf{Y}}) = 0$. The main drawback is that this strategy may underestimate the variance although it may predict the mean very well. This is a common problem for variational Bayes especially when the density model is not sufficiently accurate. We include a mutual information term in equation (4.7) to alleviate this issue. Since the joint PDF of \mathbf{Y} and \mathbf{X} will be defined as $q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}$ for sample generation, we adjust the two models $q_{\mathbf{Y}|\mathbf{X},\theta}$ and $p_{\mathbf{X},\theta}$ to minimize the uncertainty of \mathbf{Y} after \mathbf{X} is obtained, i.e., maximizing the mutual information between \mathbf{Y} and \mathbf{X} using the joint PDF induced by $q_{\mathbf{Y}|\mathbf{X},\theta}p_{\mathbf{X},\theta}$. This is possible because KRnet provides a large family of prior distributions. Numerical experiments show that maximizing the mutual information between \mathbf{X} and \mathbf{Y} is able to improve the estimation of the variance for properly chosen λ , which implies that more information from the region of low density can be kept by increasing the weight of the mutual information term. In general, we obtain the best prediction of the mean at $\lambda = \infty$, and the best prediction of the variance at a finite λ . This will be demonstrated later by numerical experiments.

5. NUMERICAL EXPERIMENTS

In this section, we examine VAE-KRnet by some numerical experiments. All algorithms are implemented by Tensorflow 2 and the optimization solver is chosen as ADAM with a learning rate 1e-3 [20]. All neural networks used in equation (2.9), encoder and decoder have fully connected hidden layers. For simplicity, the neural networks for both encoder and decoder have the same configuration. The neural networks for $f_{\text{pr}}(\cdot)$ and $f_{\text{en}}(\cdot)$ differs only with respect to the depth or the number

of the general coupling layers $f_{k,i}^{\text{inner}}$. In KRnet, the dimension will be reduced one by one if a specification is not given explicitly. No nonlinear invertible layers and rotation layers will be used. We specify some parameters: D : the number of hidden layers for both encoder and decoder, N_D : the number of neurons for each hidden layer in the encoder and decoder, L_{pr} : the number of general coupling layers in $f_{\text{pr}}(\cdot)$ (see figure 1), L_{en} : the number of general coupling layers in $f_{\text{en}}(\cdot)$, and N_L : the number of neurons for the neural network in equation (2.9). The training set has 10^5 samples. Four minibatches are used for the estimation of data distribution, and the whole training set is used for the estimation of the posterior. The validation set has $2e5$ samples whenever needed. The validation set is large such that the integration errors for the computation of statistics can be ignored compared to the errors of the model.

5.1. The linear model. We first consider the linear model (3.7), where we assume that the column vectors of \mathbf{A} are sampled from $\mathcal{N}(0, \mathbf{I})$ subject to ℓ_2 normalization. When σ is small, the distribution of \mathbf{Y} is mainly a d -dimensional distribution of $\mathbf{X} \in \mathbb{R}^d$, which is embedded in a n -dimensional space. For computation, $\mathbf{Y} \in \mathbb{R}^{10}$, and $\sigma = 0.1$.

For a prescribed prior $p_{\mathbf{X}}$, we generate the samples of \mathbf{Y} from the linear model (3.7) to form a training set. We will measure the performance of the model using the following quantity (see equation (3.1))

$$\delta_{\theta, \phi, \beta, \alpha} = \mathbb{E}_{p_{\mathbf{Y}}} [D_{\text{KL}}(q_{\mathbf{X}|\mathbf{Y}, \phi, \alpha} \| p_{\mathbf{X}|\mathbf{Y}, \theta, \beta})] = -\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\theta, \phi, \beta, \alpha}^{\mathbf{X}}(\mathbf{Y})] - h(\mathbf{Y}), \quad (5.1)$$

where $h(\mathbf{Y})$ is the differential entropy of $p_{\mathbf{Y}}$. Assuming that $p_{\mathbf{X}, \beta}$ and $p_{\mathbf{Y}|\mathbf{X}, \theta}$ cover the true prior $p_{\mathbf{X}, \text{true}}$ and the true likelihood $p_{\mathbf{Y}|\mathbf{X}, \text{true}}$, $\delta_{\theta, \phi, \beta, \alpha} = 0$ if $q_{\mathbf{X}|\mathbf{Y}, \phi, \alpha}$ is able to recover $p_{\mathbf{X}|\mathbf{Y}, \theta, \beta}$ induced by $p_{\mathbf{X}, \beta}$ and $p_{\mathbf{Y}|\mathbf{X}, \theta}$. Based on the definition of the linear model, we have

$$h(\mathbf{Y}) = -\mathbb{E}_{p_{\mathbf{Y}}} [\log p_{\mathbf{Y}}] = -\mathbb{E}_{p_{\mathbf{Y}}} [\log \mathbb{E}_{p_{\mathbf{X}}} [p_{\mathbf{Y}|\mathbf{X}}]],$$

which can be computed at the pre-processing stage. $\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\theta, \phi, \beta, \alpha}^{\mathbf{X}}(\mathbf{Y})]$ will be approximated as

$$\mathbb{E}_{p_{\mathbf{Y}}} [\mathcal{L}_{\theta, \phi, \beta, \alpha}^{\mathbf{X}}(\mathbf{Y})] \approx \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\theta, \phi, \beta, \alpha}^{\mathbf{X}}(\mathbf{y}^{(i)}),$$

where $\{\mathbf{y}^{(i)}\}_{i=1}^N$ is a validation set which is independent of the training set.

5.1.1. A Gaussian prior. Let $\mathbf{X} \in \mathbb{R}^2$,

$$p_{\mathbf{X}}(\mathbf{x}) = \mathcal{N}(0, \Sigma_{\mathbf{X}}), \quad p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2\mathbf{I}).$$

We have

$$p_{\mathbf{Y}} = \mathcal{N}(0, \sigma^2\mathbf{I} + \mathbf{A}\Sigma_{\mathbf{X}}\mathbf{A}^{\text{T}}),$$

which yields that

$$h(\mathbf{Y}) = 5(1 + \log(2\pi)) + \frac{1}{2} \log |\sigma^2\mathbf{I} + \mathbf{A}\Sigma_{\mathbf{X}}\mathbf{A}^{\text{T}}|.$$

Let $\Sigma_{\mathbf{X}} = \mathbf{I}$. We sample the two column vectors of \mathbf{A} from $\mathcal{N}(0, \mathbf{I})$ then normalize them. We know from section 3.2 that the posterior can be recovered by the canonical VAE subject to a rotation of \mathbf{X} . We let $D = 2$ and $N_D = 32$. We add one scaling and bias layer after each hidden layer, see section 2.1, to improve the efficiency.

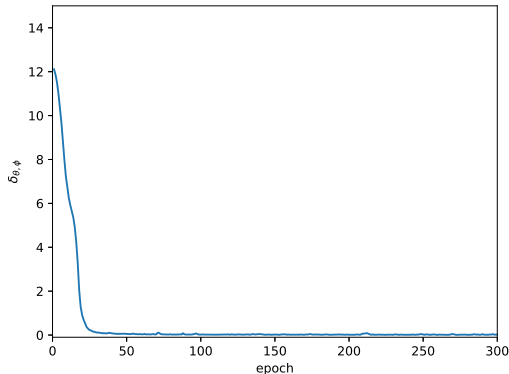


FIGURE 3. The convergence behavior of VAE for the linear model with a Gaussian prior.

The convergence behavior is shown in figure 3, where a fast decay to zero has been observed, indicating that the model has been exactly recovered.

5.1.2. *A 2d Gaussian prior with a hole.* We now look at a 2d non-Gaussian prior. We assume that $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I})$. To introduce correlation between X_1 and X_2 , we consider the data satisfying

$$B = \{\mathbf{x} \mid \|\mathbf{R}^{\alpha, \theta} \mathbf{x}\|_2 \geq C\},$$

where $0 < C < \infty$, and \mathbf{R} is a matrix defined as

$$\mathbf{R}^{\alpha, \theta} = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix},$$

corresponding to a rotation and a stretch. Simply speaking, the distribution is given by 2d standard Gaussian subject to an elliptic hole. We let $\alpha = 3.0$ and $\theta = \pi/4$. To this end, we have prescribed

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{I_B(\mathbf{x})p_{X_1}(x_1)p_{X_2}(x_2)}{\int_{\mathbb{R}^2} I_B(\mathbf{x})p_{X_1}(x_1)p_{X_2}(x_2)dx_1dx_2}, \quad p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2\mathbf{I}),$$

where $p_{X_i} = \mathcal{N}(0, 1)$ and $I_B(\mathbf{x})$ is an indicator function.

We first show the effect of the generalized prior and posterior. We consider three models: canonical VAE, VAE-KRnet I, and VAE-KRnet II, where VAE-Krnet I has a generalized prior and VAE-KRnet II has both generalized prior and posterior. For the sake of comparison we consider simple configurations. We let $D = 1$, $L_{\text{pr}} = L_{\text{en}} = 2$, $N_D = 32$, and $N_L = 24$, whenever the corresponding components are needed in the model. The convergence behavior of these three models has been shown in figure 4. It is seen that both the generalized prior and posterior are able to improve the performance of the canonical VAE. It appears that $f_{\text{pr}}(\cdot)$ can improve the performance more effectively than $f_{\text{en}}(\cdot)$.

We now compare the simulated distributions of \mathbf{Y} given by canonical VAE and VAE-KRnet II, where we let $D = 2$, $L_{\text{pr}} = 8$, $L_{\text{en}} = 2$, $N_D = 32$ and $N_L = 24$. The results have been shown in figure 5. It is seen that canonical VAE is effective to capture the main structure of the distribution while VAE-KRnet is able to capture

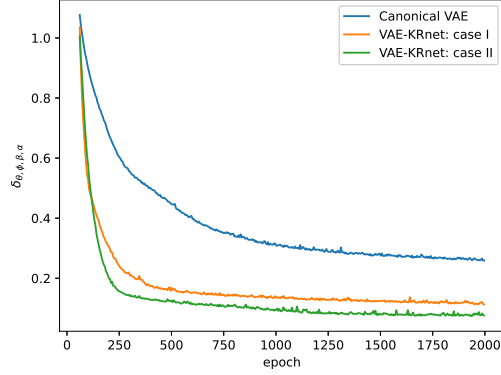


FIGURE 4. The convergence behavior of VAE and VAE-KRnet for the linear model with a non-Gaussian prior. VAE-KRnet I has a generalized prior, and VAE-KRnet II has both generalized prior and posterior.

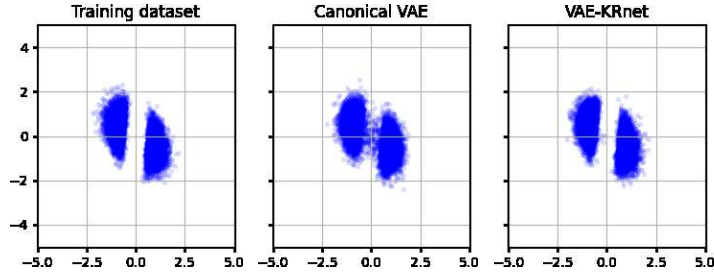


FIGURE 5. The distribution of (Y_6, Y_9) . Left: training set; Middle: samples generated by the canonical VAE; Right: samples generated by VAE-KRnet.

more details than canonical VAE. In figure 6, we compare the given prior and the learned prior by VAE-KRnet. It is seen that the learned prior shares some similarities with the given prior. Note that any invertible mapping of \mathbf{X} provides a latent variable. Therefore we do not expect the learned prior is the same as the prescribed one.

5.1.3. *A 3d Gaussian prior with holes.* We now consider a case that $\mathbf{X} \in \mathbb{R}^3$. For $\mathbf{x} = [x_1, x_2, x_3]^\top$, we let $\mathbf{x}_i = [x_i, x_{i+1}]^\top$, $i = 1, 2$, which includes two adjacent components of \mathbf{x} . We sample from $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I})$ and keep the data

$$B = \{\mathbf{x}_i \mid \|\mathbf{R}^{\alpha, \theta_i} \mathbf{x}_i\|_2 \geq C\}, \quad i = 1, 2.$$

In other words, for any two adjacent dimensions we generate an elliptic hole. We let $\alpha = 3$, $\theta_1 = \pi/4$ and $\theta_2 = 3\pi/4$ (see the top two plots in figure 9). We first check the performance of canonical VAE and VAE-KRnet with respect to D , the depth of the neural networks for the encoder and decoder. We let $L_{\text{pr}} = 8$, $L_{\text{en}} = 2$, $N_L = 24$

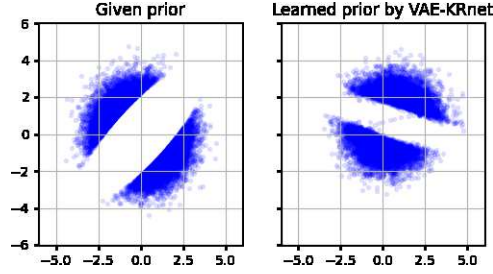


FIGURE 6. Left: the given prior distribution; Right: the learned prior distribution by VAE-KRnet.

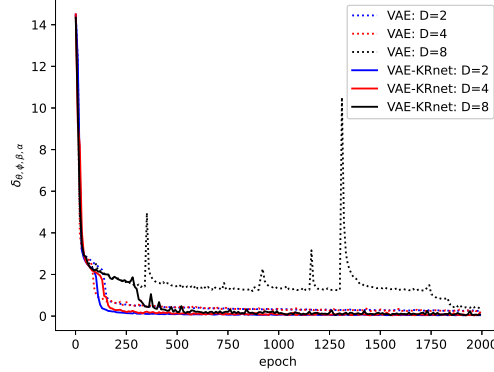


FIGURE 7. The convergence behavior of VAE and VAE-KRnet in terms of D .

and $N_D = 32$. The convergence behavior has been plotted in figure 7. First of all, VAE-KRnet has a better performance. For a fixed D , VAE-KRnet reaches a smaller loss than VAE. Second, VAE-KRnet is more robust than VAE. When $D = 8$, VAE has been stuck in a local minimizer until epoch ≈ 1750 before it goes to a better local minimizer. The introduction of generalized prior and posterior makes it much easier to escape the basin of attraction of such a local minimizer. It is seen that VAE-KRnet with $D = 8$ is able to achieve the same loss as other configurations when the epoch is about 500 although the degree of fluctuation is bigger due to the increased model complexity. In figure 8, we compare the distributions simulated by VAE and VAE-KRnet. It is seen that much more details can be captured by VAE-KRnet than VAE. In figure 9, we plotted the prior distributions given by VAE-KRnet, where the only difference in configuration is that $L_{pr} = 8, 10$. It is seen that the learned prior distributions are quite different although the two configurations of VAE-KRnet are similar and yield almost the same approximation of the data distribution.

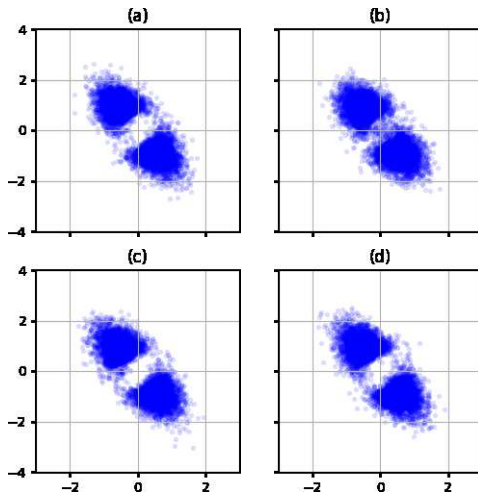


FIGURE 8. The samples of (Y_6, Y_8) given by VAE and VAE-KRnet with $D = 2$. (a): training set; (b): canonical VAE; (c): VAE-KRnet with $L_{\text{pr}} = 8$; (d): VAE-KRnet with $L_{\text{pr}} = 10$.

5.2. A linear Bayesian inverse problem. We consider the following linear model for the inverse problem:

$$\hat{\mathbf{Y}} = \mathbf{K}\mathbf{Y} + \boldsymbol{\xi}, \quad (5.2)$$

where $\mathbf{K} \in \mathbb{R}^{k \times n}$, and $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. In particular, we assume that \mathbf{K} is ill-conditioned in the sense that its singular values decays fast. Assume a Gaussian prior $\mathcal{N}(\boldsymbol{\mu}_{\text{pr}}, \boldsymbol{\Sigma}_{\text{pr}})$ is used. The posterior is

$$p_{\text{post}}(\mathbf{y}) \propto \hat{p}_{\text{post}}(\mathbf{y}) = \exp\left(-\frac{|\hat{\mathbf{y}} - \mathbf{K}\mathbf{y}|^2}{2\sigma^2}\right) \exp\left(-\frac{1}{2}|\mathbf{y} - \boldsymbol{\mu}_{\text{pr}}|_{\boldsymbol{\Sigma}_{\text{pr}}^{-1}}^2\right), \quad (5.3)$$

what $\hat{p}_{\text{post}}(\mathbf{y})$ is the unnormalized posterior such that $p_{\text{post}}(\mathbf{y}) \int \hat{p}_{\text{post}}(\mathbf{y}) d\mathbf{y} = \hat{p}_{\text{post}}(\mathbf{y})$, and $|\mathbf{y}|_{\boldsymbol{\Sigma}_{\text{pr}}^{-1}}^2 = \mathbf{y}^T \boldsymbol{\Sigma}_{\text{pr}}^{-1} \mathbf{y}$ defines a weighted ℓ_2 norm induced by the precision matrix. We want to find a low-dimension latent random variable $\mathbf{X} \in \mathbb{R}^d$ for \mathbf{Y} such that $\int p_{\mathbf{X}, \mathbf{Y}} d\mathbf{x} \approx p_{\text{post}}(\mathbf{y})$, where $p_{\mathbf{X}, \mathbf{Y}}$ is the joint PDF of \mathbf{X} and \mathbf{Y} , and will be provided by VAE-KRnet.

We define problem (5.2) using an integral equation

$$g(x) = \int_{\mathbb{R}} K(x, y) f(y) dy, \quad (5.4)$$

where $K(x, y)$ is the kernel of a compact operator that is of trace class, positive and self-adjoint. Let $(\lambda_i, e_i(x))$ indicate the eigen-pairs of $K(x, y)$. Assume that

$$f(y) \approx \sum_{i=1}^M f_i e_i(y). \quad (5.5)$$

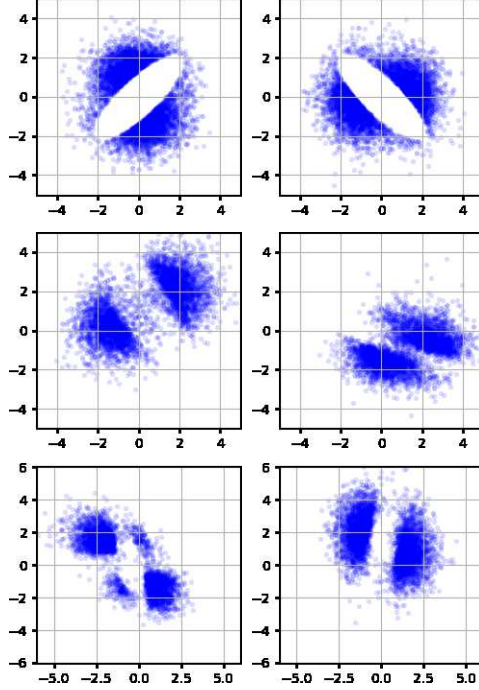


FIGURE 9. The prescribed and learned prior distributions. For each row, the left plot gives the distribution of (X_1, X_2) , and the right plot corresponds to (X_2, X_3) . From top to bottom, the first row corresponds to the prescribed prior distribution, and the second and third rows correspond to the learned prior distributions given by VAE-KRnet with the same configuration except that $L_{pr} = 8, 10$ respectively.

We consider the equation

$$g(x_j) \approx \sum_{i=1}^M f_i \lambda_i e_i(x_j), \quad j = 1, \dots, N_x, \quad (5.6)$$

where x_i are collocation points. This yields a linear system

$$\mathbf{g} = \mathbf{E}\mathbf{\Lambda}\mathbf{f}, \quad (5.7)$$

where $e_{ij} = e_i(x_j)$, $g_j = g(x_j)$, $i = 1, \dots, M$, $j = 1, \dots, N_x$. We then let $\mathbf{K} = \mathbf{E}\mathbf{\Lambda}$ in equation (5.2) and $\hat{\mathbf{y}} = \mathbf{g}$ is the data. We need to infer the coefficients f_i in equation (5.5).

For simplicity and without loss of generality, we here consider an artificial case, where we let $e_i(x) = \frac{1}{\sqrt{\pi}} \cos(ix)$ with $x \in [0, 2\pi]$, and $\lambda_i = i^{-\gamma}$ with $\gamma > 0$. The

collocation points are sampled from a uniform distribution on $[0, 2\pi]$. This way, the column vectors of \mathbf{E} are nearly mutually orthogonal due to the properties of $\cos(ix)$. The condition number of $\mathbf{K}^\top \mathbf{K}$ depends on the value of γ , where the eigenvalue λ_i^2 decays faster for a larger γ . We define a vector \mathbf{y}_0 with $y_{0,i} = i^{-2.0} \sin(i)$, $i = 1, \dots, n$, and generate the data $\hat{\mathbf{y}} = \mathbf{K}\mathbf{y}_0 + \sigma\boldsymbol{\xi}_0$, where $\boldsymbol{\xi}_0$ is a sample from $\mathcal{N}(0, \mathbf{I})$. We then consider a Bayesian inverse problem (5.2) using $\hat{\mathbf{y}}$ as the given data. The true posterior is

$$p_{\text{post}}(\mathbf{y}) = \mathcal{N}(\sigma^{-2}(\boldsymbol{\Sigma}_{\text{pr}}^{-1} + \sigma^{-2}\mathbf{K}^\top \mathbf{K})^{-1}\mathbf{K}^\top \hat{\mathbf{y}}, (\boldsymbol{\Sigma}_{\text{pr}}^{-1} + \sigma^{-2}\mathbf{K}^\top \mathbf{K})^{-1}). \quad (5.8)$$

Since we often choose $\boldsymbol{\Sigma}_{\text{pr}}$ as a diagonal matrix, the covariance matrix of $p_{\text{post}}(\mathbf{y})$ is nearly diagonal by the definition of $\mathbf{K} = \mathbf{E}\boldsymbol{\Lambda}$, which implies that the components of \mathbf{Y} are nearly independent. To consider dimension reduction, correlation should be introduced. We define a matrix $b_{ij} = e^{-|i-j|/\alpha}$ with $\alpha > 0$ and $i, j = 1, \dots, n$, and redefine $\mathbf{K} = \mathbf{E}\mathbf{B}\boldsymbol{\Lambda}$. The parameter α acts as a correlation length. Note that the column vectors of $\mathbf{E}\mathbf{B}$ are not nearly mutually orthogonal any more. Letting $\mathbf{y} = \mathbf{0}$, the normalization constant for \hat{p}_{post} can be computed as

$$C = \frac{\hat{p}_{\text{post}}(\mathbf{0})}{p_{\text{post}}(\mathbf{0})} = \frac{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_{\text{post}}|}}{\exp\left(\frac{1}{2\sigma^2} |\hat{\mathbf{y}}|^2 + \frac{1}{2} |\boldsymbol{\mu}_{\text{pr}}|_{\boldsymbol{\Sigma}_{\text{pr}}^{-1}}^2 - \frac{1}{2} |\boldsymbol{\mu}_{\text{post}}|_{\boldsymbol{\Sigma}_{\text{post}}^{-1}}^2\right)}, \quad (5.9)$$

where $\boldsymbol{\mu}_{\text{post}}$ and $\boldsymbol{\Sigma}_{\text{post}}$ are the mean and covariance matrix of the posterior (5.8).

We consider several PDF models for the approximation of the posterior. 1) The mean-field variational family, where all random variables are assumed to be mutually independent. For our problem, each dimension will be assumed to be Gaussian; 2) KRnet; and 3) VAE-KRnet. The mean-field variational family is widely used in practice due to its simplicity and efficiency. We do not include VAE here since VAE-KRnet is more robust than VAE. For the mean-field model, we simply use the ADAM method for optimization without taking advantage of the mutual independence like the CAVI algorithm [2]. A direct generalization of the mean-field variational model is the mixture of Gaussians, which is not included here. Instead, we consider VAE-KRnet as a generalization of the mixture of Gaussians, since the latent variables for VAE-KRnet is much more general than the latent variable of the Gaussian mixture model.

Another issue is the computation of statistics. The statistics will be computed using the model that yields the minimum loss with respect to the validation set. Actually, for our experiments the training set is large enough, meaning that we do not observe that the error will increase in terms of the validation set after the optimization iteration has stabilized.

We let $\sigma = 0.05$ and $\mathcal{N}(\boldsymbol{\mu}_{\text{pr}}, \boldsymbol{\Sigma}_{\text{pr}}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_{\text{pr}})$, where $\boldsymbol{\Lambda}_{\text{pr}}$ is a diagonal matrix with $\lambda_{\text{pr},i} = i^{-2.5}$, $i = 1, \dots, n$. For matrix \mathbf{B} , we let $b_{ij} = e^{-|i-j|/3.0}$, $i, j = 1, \dots, n$. Note that from equation (4.6) we have

$$\mathcal{D}_{\boldsymbol{\theta}}^{\text{pdf}}(\hat{p}_{\mathbf{Y}}) \geq -\log C.$$

The lower bound can be computed by equation (5.9). We will consider two cases when $n = 10, 50$. For KRnet, we let $L = 6$, $K = 5$, $N_L = 24$ for both $n = 10$ and $n = 50$. The dimensions will be deactivated by two if $n = 10$ and by ten if $n = 50$. In other words, the model complexity of KRnet is the same for $n = 10, 50$. For VAE-KRnet, we let $L_{\text{pr}} = 6$, $L_{\text{en}} = 2$, $N_L = 24$, $N_D = 32$ for all cases. In $f_{\text{pr}}(\cdot)$ and $f_{\text{en}}(\cdot)$, the dimensions will be deactivated by two. Let $r(x; \mathbf{Y}) = \sum_{i=1}^n Y_i e_i(x)$.

After we approximate the posterior of \mathbf{Y} , we compute the $\mathbb{E}[r](x)$ and $\text{Var}(r)(x)$ and compare them to the exact values.

We first consider a 10-dimensional case, where we have $-\log C = 107.94$ from equation (5.9). In figure 10 we plotted the evolution behavior of the ADAM method, where the global behavior is given on the left with respect to the loss, and the stabilized behavior is given on the right with respect to the relative error of the loss:

$$\frac{|\mathcal{D}_{\theta}^{\text{pdf}}(\hat{p}_{\mathbf{Y}}) + \log C|}{|\log C|}.$$

Interestingly, all VAE-KRnet models decay much faster than the mean-field variational model although they are much more complicated. Due to the correlation introduced by \mathbf{B} , the mean-field variational model becomes stabilized at a larger relative error than other PDF models, where KRnet performs the best and VAE-KRnet yields a smaller loss for a larger λ . After the iteration number reaches $3e5$, we compute the minimum loss within every 1000 iteration with respect to the validation set, and the results are given in the right plot of figure 10. It is seen that the minimum loss is quite steady although a lot of fluctuations exist in the optimization iteration.

In figure 11, we plotted the predicted mean and variance. It is seen that the mean is well predicted by all models while the prediction of the variance varies significantly. KRnet yields the best approximation. The mean-field variational model barely captures any characteristics of the variance. VAE-KRnet with $\lambda = 2$ yields a better estimation of the variance than $\lambda = \infty$. In figure 12 we plotted the effect of the dimension of the latent variable on the left and the effect of the value of λ on the right. It is seen that the prediction has been improved by increasing d , which is expected. To show the effect of λ , we compute the errors as follows. Let $\hat{r}(x; \hat{\mathbf{Y}})$ be an approximation of $r(x; \mathbf{Y})$. We check the following errors:

$$\frac{\|\mathbb{E}[\hat{r}] - \mathbb{E}[r]\|_{L_2}}{\|\mathbb{E}[r](x)\|_{L_2}}, \quad \frac{\|\text{Var}^{1/2}(\hat{r}) - \text{Var}^{1/2}(r)\|_{L_2}}{\|\mathbb{E}[r](x)\|_{L_2}},$$

for the mean and the standard variation respectively, where the $\|\cdot\|_{L_2}$ is with respect to the spacial variable x . We are interested in a range of λ , in which the mutual information term in equation (4.7) helps improve the prediction of the standard deviation. As discussed in section 4.3.2, when $\lambda \leq 1$ the optimization problem (4.7) becomes ill-posed. For the problem studied, a smaller error of the standard deviation is observed when we increase λ to about 2. As λ continues to increase, the error of the standard deviation has an overall trend to decrease until λ is about 3, after which we expect that the effect of the mutual information term will become more and more weaker and the error of the standard deviation will increase and approach the error given by $\lambda = \infty$, i.e., the blue horizontal line. It is seen that VAE-KRnet with $\lambda = \infty$ yields a much smaller error for the mean (the red line) than for the standard deviation (the blue line). Within quite a wide range of λ , VAE-KRnet with a finite λ yields a much more accurate estimation of the variance than VAE-KRnet with $\lambda = \infty$. However, VAE-KRnet with a finite λ yields a worse estimation of the mean than VAE-KRnet with $\lambda = \infty$.

We subsequently look at the case that $n = 50$, where we change α in $b_{ij} = e^{-|i-j|/\alpha}$ from 3 to 10. So more correlations can be introduced such that a relatively small number of latent random variables is needed. For this case, $-\log C = 632.11$.

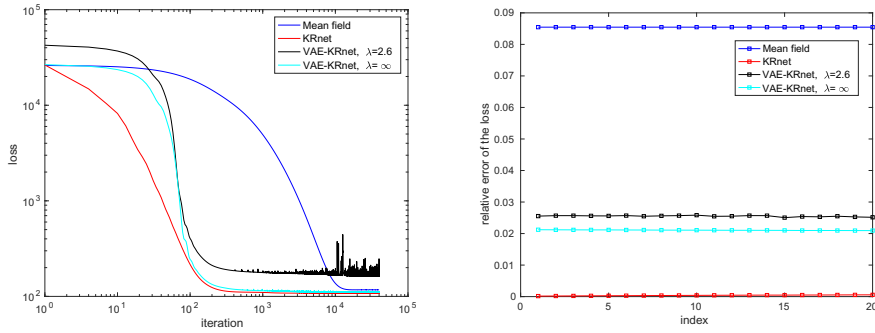


FIGURE 10. The evolution behavior of the ADAM method for different PDF models. $n = 10$. Left: iterations up to $3e5$; Right: iterations from $3e5$ to $5e5$. Each node corresponds to the minimum loss in every 1000 iterations.

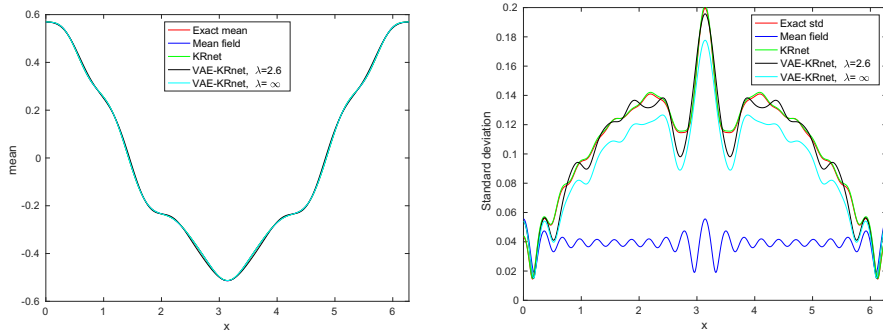


FIGURE 11. The statistics given by different PDF models. $n = 10$. $d = 4$ for VAE-KRnet. Left: Mean; Right: Variance.

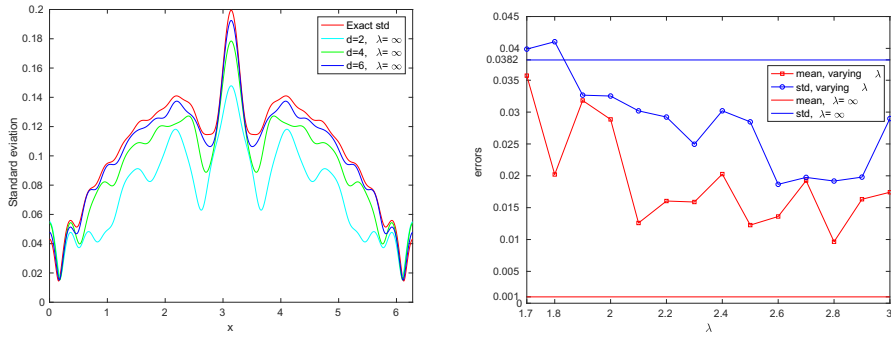


FIGURE 12. VAE-KRnet for different d and λ . Left: varying d with $\lambda = \infty$. Right: varying λ with $d = 4$.

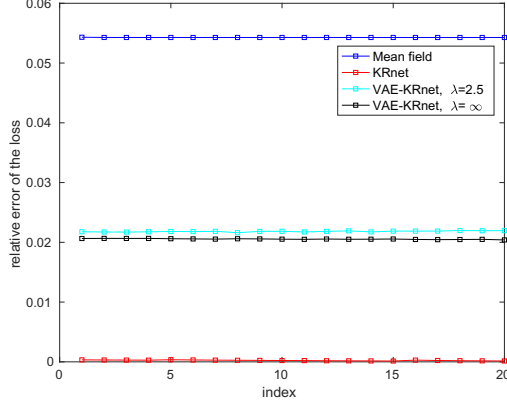


FIGURE 13. The stabilized iterations of the ADAM method for different PDF models. $n = 50$. $d = 8$ for VAE-KRnet. Iterations from $3e5$ to $5e5$. Each node corresponds to the minimum loss in every 1000 iterations.

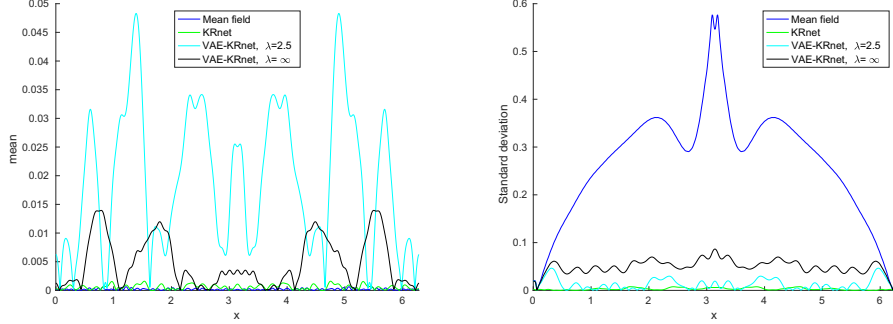


FIGURE 14. Errors of different models on the validation set. The errors are scaled by $\|\mathbb{E}[r]\|_{L_2}$. $n = 50$. $d = 8$. Left: mean; Right: variance.

We let $d = 8$. The global evolution behavior of the ADAM method is similar to the case that $n = 10$, see figure 13. Note that although the number of dimensions is relatively large, the KRnet accurately captured the correlations using a model that has the same complexity (in terms of K , L and N_L) as the model for the case $n = 10$. We plot pointwise errors in terms of x in figure 14 for the mean on the left and for the standard deviation on the right. Both errors are scaled by the L_2 norm of the exact mean, i.e., $\|\mathbb{E}[r]\|_{L_2}$. For this case, the mean-field model yields the best estimation for the mean but no useful estimation for the standard deviation. KRnet yields accurate predictions for both the mean and the standard deviation. VAE-KRnet with $\lambda = \infty$ yields a better estimation for the mean and a worse estimation for the variance than VAE-KRnet with $\lambda = 2.5$. The rank for

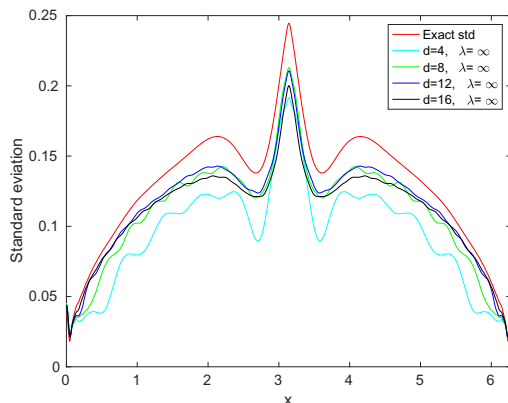


FIGURE 15. Standard deviation given by VAE-KRnet with varying d . $n = 50$.

predicting the mean is: mean-field model, KRnet, VAE-KRnet with $\lambda = \infty$, VAE-KRnet with $\lambda = 2.5$. The rank for predicting the variance is: KRnet, VAE-KRnet with $\lambda = 2.5$, VAE-KRnet with $\lambda = \infty$ and mean-field model.

We finally compare the predictions given by VAE-KRnet in terms of the dimension of the latent variables. The results are plotted in figure 15. For a certain d , we choose $K = d/4$ for KRnet, i.e., the dimensions will be deactivated by $d/4$. This way, the overall number of model parameters of VAE-KRnet remains almost the same for a varying d . It is seen that as d increases the improvement on the prediction will cease at a certain d . The reason is twofold: first, VAE does not converge to the full model when d increases to n because of the model error; second, we constrained the model complexity with a roughly constant number of model parameters.

5.3. One-dimensional elliptic problem. We now consider a one-dimensional elliptic problem

$$-\frac{d}{dx} \left(e^{a(x;\omega)} \frac{du}{dx} \right) = 1, \quad x \in [0, 1], \quad (5.10)$$

with homogeneous boundary conditions, where $a(x; \omega)$ is a Gaussian field. We associate $a(x; \omega)$ with a Gaussian measure $\mu = \mathcal{N}(0, \mathcal{C})$ with zero mean and covariance operator \mathcal{C} in $L^2([0, 1])$. The covariance operator \mathcal{C} is chosen as

$$\mathcal{C} = \alpha^{-1}(I - \Delta)^{-s} = \alpha^{-1}\mathcal{A}^{-s}, \quad (5.11)$$

where $\alpha > 0$ is a constant, $s > \frac{1}{2}$, and the domain of \mathcal{A} is

$$D_{\mathcal{A}} = \{a(x) \in H^2([0, 1]) : \partial_x a = 0 \text{ at } x = 0, 1\}. \quad (5.12)$$

We infer the coefficient $a(x)$ using pointwise observations of $u(x)$:

$$\hat{u}_i = u(x_i) + \eta_i, \quad i = 1, \dots, J, \quad (5.13)$$

where J is the total number of observation locations x_i and $\eta_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. Gaussian random variables with zero mean and variance σ^2 . In the framework

of Bayesian inverse problem, the posterior measure ν can be represented with the Radon-Nikodym derivative

$$\frac{d\nu}{d\mu}(a(x)|\hat{\mathbf{u}}) \propto \exp(-\Phi(a, \hat{\mathbf{u}})), \quad (5.14)$$

where

$$\Phi(a, \hat{\mathbf{u}}) = \frac{1}{2\sigma^2} \sum_{i=1}^K (\hat{u}_i - u(x_i))^2 \quad (5.15)$$

is the potential defined by the distribution of the observational noise η_i .

We will approximate the posterior distribution using VAE-KRnet. We discretize the problem using linear finite elements given by $\text{span}\{\theta_i(x)\}_{i=1}^M$. Let \mathbf{A} be the representation of \mathcal{A} in the finite element space, i.e.,

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{K} + \mathbf{I}, \quad (5.16)$$

where \mathbf{M} is the mass matrix and \mathbf{K} is the stiffness matrix. Let (σ_i, \mathbf{v}_i) be the eigenpairs of \mathbf{A} , i.e.,

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{v}_i, \quad \mathbf{A}\mathbf{V} = \mathbf{V}\boldsymbol{\Sigma}, \quad \mathbf{v}_i^\top \mathbf{M}\mathbf{v}_j = \delta_{ij}, \quad (5.17)$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix with nonzero entries σ_i and \mathbf{A} include all eigenvectors \mathbf{v}_i . Using the matrix transfer technique, we may define the discrete representation of \mathcal{A}^{-s} as

$$\mathbf{A}^{-s} = \mathbf{V}\boldsymbol{\Sigma}^{-s}\mathbf{V}^{-1}, \quad (5.18)$$

where

$$\mathbf{A}^{-s}\mathbf{v}_i = \sigma_i^{-s}\mathbf{v}_i = \lambda_i\mathbf{v}_i. \quad (5.19)$$

We then have the finite element representation of $a(x) = \mathbf{a}^\top \boldsymbol{\Theta}(x)$ using the following Karhunen-Loeve expansion

$$\mathbf{a} = \mathbf{a}_0 + \mathbf{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{\gamma}, \quad (5.20)$$

where $\boldsymbol{\gamma} \in \mathcal{N}(0, \mathbf{I})$, $\mathbf{a} \in \mathbb{R}^M$ includes the coefficients of the finite element approximation, $\boldsymbol{\Theta}(x) \in \mathbb{R}^M$ includes all finite element basis functions and \mathbf{a}_0 is the mean. With respect to \mathbf{a} , we have the discretized posterior distribution

$$\pi(\mathbf{a}|\hat{\mathbf{u}}) \propto \exp(-\Phi(\mathbf{a}, \hat{\mathbf{u}})) \mathcal{N}(\mathbf{a}_0, \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top), \quad (5.21)$$

where $\Phi(\mathbf{a}, \hat{\mathbf{u}})$ is the discretized potential induced by \mathbf{a} . Note that the dimension of \mathbf{a} is the number of finite element basis functions.

For this case explicit forms are not available for the posterior measure ν or its discrete version $\pi(\mathbf{a}|\hat{\mathbf{u}})$. We here use the function-space hybrid Monte Carlo (HMC) to generate some reference solutions [4]. We use $M = 128$ equidistant linear finite elements to discretize $[0, 1]$. The observations are generated with $a(x) = \exp(0.1 \cos(2\pi x))$ on locations $x_i = \frac{i}{64}$, $i = 1, \dots, 63$, where the noise amplitude σ is 5% of the maximum of $u(x)$. We let $\lambda = 2.8$ in equation (4.7) according to the right plot of figure 12. The dimension of the latent space is set to be $d = 16$ in contrast to the dimension $n = M + 1 = 129$ of \mathbf{a} .

We use the same configuration as the case in the previous section for the definition of VAE-KRnet. The only difference is that in KRnet the number of active dimensions will be deactivated by $16/4 = 4$. In each optimization step, we need to compute $\nabla_{\mathbf{a}}\pi$, where one forward problem and one adjoint problem are solved, and other gradients can be done by the automatic differentiation in Tensorflow. Due to the cost of solving two elliptic problems in each iteration, we consider a relatively small training set of size 1024 with a batch size 128. In contrast to the

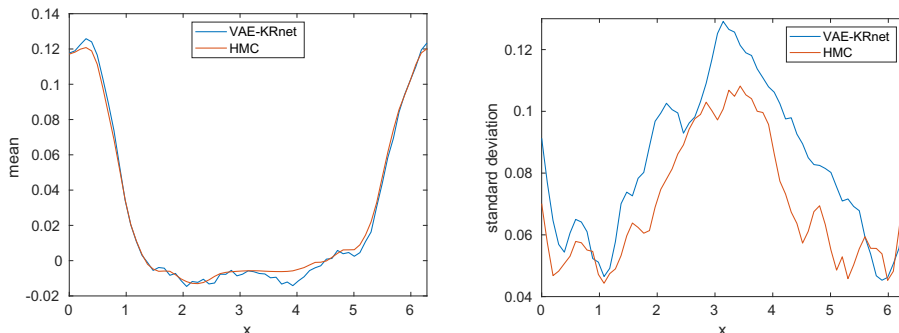


FIGURE 16. Statistics of \mathbf{a} given by VAE-KRnet and HMC for the elliptic problem. $n = 129$. $d = 16$. Left: mean; Right: standard deviation.

hybrid Monte Carlo, the computation of the forward and adjoint problems can be parallelized when training VAE-KRnet. We train the model for 600 epochs and use the models given by the last 60 epochs for a time average when computing predictions. For the hybrid Monte Carlo, we generated 10^5 samples. For this case, the time consumption for VAE-KRnet is about one fourth of that for HMC.

In figure 16 we plotted some statistics of \mathbf{a} in terms of the posterior distribution. It is seen that VAE-KRnet provides consistent results with HMC for both the mean and the standard deviation although the size of the training set is relatively small. We here simply use the one-dimensional elliptic problem to verify the effectiveness of VAE-KRnet. Many computational issues remain to improve the efficiency of VAE-KRnet, which are beyond the scope of this paper.

6. SUMMARY

In this paper, we have developed a family of probability density models by coupling VAE and KRnet. KRnet is an effective invertible transport map, and VAE is an effective technique for dimension reduction. VAE-KRnet inherits the advantages of both VAE and KRnet. For a linear system, the encoder of the canonical VAE mainly does the following things (see Remark 3.2): 1) A mapping from a standard Gaussian to an arbitrary distribution in the latent space; 2) A rotation to make the encoder diagonal; and 3) computation of the covariance matrix; In the canonical VAE, all these tasks are achieved by the encoder. In VAE-KRnet, the first task is achieved by a KRnet for an arbitrary prior, and the second task is achieved by another KRnet for the correlation between dimensions. Compared to the canonical VAE, each component of VAE-KRnet has a more specific task, which improves both the performance and the robustness. We applied VAE-KRnet to variational Bayes to approximate the posterior. VAE-KRnet has demonstrated some promising potentials: 1) It covers a wide range of data dimensions by varying the number of dimensions of the latent space from zero (KRnet) to d (VAE-KRnet). 2) By taking into account the mutual information, a possibility is provided to improve the underestimation of the variance by varying the parameter λ , which yields a statistics-oriented way for model selection. 3) Increasing the dimensionality d will improve the approximation. Of course, d would be limited by the model capability

of both VAE and KRnet as shown in figure 15. However, varying d does not need to introduce a significant change of the model complexity. 4) For linear Bayesian inverse problems, the KRnet performs very well for high-dimensional cases. It is seen that KRnet yields the best prediction for the last example with $n = 50$. On one hand, this is because the posterior is Gaussian, which is relatively simple; on the other hand, it demonstrates the modeling capability of KRnet. 5) The numerical experiments on the elliptic problem demonstrate that VAE-KRnet is able to produce consistent results with HMC using a relatively small training set. The current results are very encouraging for us to apply VAE-KRnet to model the posterior of nonlinear Bayesian inverse problems.

ACKNOWLEDGMENT

X. Wan has been supported by NSF grant DMS-1913163, and S. Wei has been supported by NSF grant ECCS-1642991.

REFERENCES

- [1] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein GAN*, (2017), arXiv:1701.07875v3.
- [2] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Variational inference: A review for statisticians*, (2018), arXiv:1601.00670v9.
- [3] R. van den Berg, L. Hasenclever, J. M. Tomczak and M. Welling, *Sylvester normalizing flows for variational inference*, (2019), arXiv:1601.00670v9.
- [4] A. Beskos, F. J. Pinski, J. M. Sanz-Serna and A. M. Stuart, *Hybrid Monte Carlo on Hilbert spaces*, Stochastic Processes and their Applications, 121 (2019), pp. 2201–2230.
- [5] G. Carlier, A. Galichon, and F. Santambrogio, *From Knothe’s transport to Brenier’s map and a continuation method for optimal transport*, SIAM J. Math. Anal., 41(6) (2010), pp. 2554–2576.
- [6] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations*, (2019), arXiv:1806.07366v5.
- [7] L. Dinh, D. Krueger, and S. Bengio, *Nice: non-linear independent components estimation*, (2014), arXiv:1410.8516.
- [8] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density estimation using real NVP*, (2017), arXiv:1605.08803v3.
- [9] E. Dupont, A. Doucet, and Y. W. Teh, *Augmented neural ODEs*, (2019), arXiv:1904.01681v3.
- [10] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Birkhäuser, 2010.
- [11] C. Finlay, J.-H. Jacobsen, L. Nurbekyan, and A. M. Oberman *How to train your neural ODE: the world of Jacobian and kinetic regularization*, (2020), arXiv:2002.02798v3.
- [12] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numerica, (2015), pp. 259–328.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in Neural Information Processing Systems, (2014), 2672–2680.
- [14] A. Graves, *Generating sequences with recurrent neural networks*, (2013), arXiv:1308.0850.
- [15] A. Grover, M. Dhar, and S. Ermon, *Flow-GAN: Combining maximum likelihood and adversarial learning in generative models*, (2018), arXiv:1705.08868v2.
- [16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017.
- [17] S. Ioffe, and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariance shift*, (2015), arXiv:1502.03167v3.
- [18] D. P. Kingma, and M. Welling, *Auto-encoding variational Bayes*, (2014), arXiv:1312.6114v10.
- [19] D. P. Kingma, and P. Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions*, (2018), arXiv:1807.03039v2.

- [20] D. P. Kingma, and J. L. Ba, *ADAM: A method for stochastic optimization*, (2017), arXiv:1412.6980v9.
- [21] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, *Improving variational inference with inverse autoregressive flow*, Advances in Neural Information Processing Systems, (2016), pp. 4743–4751.
- [22] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, *Pixel recurrent neural networks*, (2016), arXiv:1601.06759.
- [23] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, *Conditional image generation with PixelCNN decoders*, (2016), arXiv:1606.05328.
- [24] G. Papamakarios, T. Pavlakou, and I. Murray, *Masked autoregressive flow for density estimation*, (2018), arXiv:1705.07057v4.
- [25] D. Rezende, and S. Mohamed, *Variational inference with normalizing flows*, ICML, (2015), 1530–1538.
- [26] R. Y. Rubinstein, and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, Springer-Verlag, New York, 2004.
- [27] D. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, 2nd Edition, John Wiley & Sons, Inc., 2015.
- [28] A. Spatini, D. Bigoni, and Y. Marzouk, *Inference via low-dimensional couplings*, (2017), arXiv:1703.06131v4.
- [29] K Tang, X. Wan, and Q. Liao, *Deep density estimation via invertible block-triangular mapping*, Theoretical & Applied Mechanics Letters, **10**, 2020, 000-5.
- [30] K Tang, X. Wan, and Q. Liao, *An adaptive deep density estimation for Fokker-Planck equations*, submitted.
- [31] X. Wan, and S. Wei, *Coupling the reduced-order model and the generative model for an importance sampling estimator*, J. Compt. Phys., in press.
- [32] L. Yang, and G. E. Karniadakis, *Potential flow generator with L_2 optimal transport regularity for generative models*, (2019), arXiv:1908.11462v1.
- [33] L. Zhang, W. E, and L. Wang, *Monge-Ampère flow for generative modeling*, (2018), arXiv:1809.10188v1.
- [34] J. Zhu, D. Zhao, and B. Zhang, *LIA: Latently Invertible Autoencoder with Adversarial Learning*, (2019), arXiv:1906.08090v1.

Current address, X. Wan: Department of Mathematics, and Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803

Email address, X. Wan: [xlwan@lsu.edu](mailto:xlw@lsu.edu)

Current address, S. Wei: Division of Electrical & Computer Engineering, Louisiana State University, Baton Rouge, LA 70803

Email address, S. Wei: swei@lsu.edu