VAE-KRNET AND ITS APPLICATIONS TO VARIATIONAL BAYES

XIAOLIANG WAN AND SHUANGQING WEI

ABSTRACT. In this work, we have proposed a generative model for density estimation, called VAE-KRnet, 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 variables. Using a linear model between the data and the latent variables, we show that VAE-KRnet can be more effective and robust than the canonical VAE. As an application, we apply VAE-KRnet to variational Bayes to approximate the posterior. The variational Bayes approaches are usually based on the minimization of the Kullback-Leibler (KL) divergence between the model and the posterior, which often underestimates the variance if the model capability is not sufficiently strong. However, for high-dimensional distributions, it is very challenging to construct an accurate model since extra assumptions are often needed for efficiency, e.g., the mean-field approach assumes mutual independence between dimensions. When the number of dimensions is relatively small, KRnet can be used to approximate the posterior effectively with respect to the original random variable. For high-dimensional cases, we consider VAE-KRnet to incorporate with the dimension reduction. To alleviate the underestimation of the variance, we include the maximization of the mutual information between the latent random variable and the original one when seeking an approximate distribution with respect to the KL divergence. 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 [20]. 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 [9, 16, 17, 18], variational autoencoders (VAE) [12, 15], and flow-based generative models [4, 5, 13, 24]. A particular case is the generative adversarial networks (GANs) [8, 1], which requires finding a Nash equilibrium of a game. Recently, the coupling of different modeling strategies has also been explored. The flow-based model was coupled with GAN in [10] to obtain a likelihood for GAN; The VAE, flow-based model and GAN were coupled in [25] for more flexibility and efficiency. The main goal of deep generative models is to generate new data that are consistent with the 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 standard Gaussian to the desired data distribution without using the likelihood. Although the main goal is not density estimation, many of the deep generative models produce a density model,
e.g., the flow-based models actually define an invertible transport map between two probability measures. 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 nonlinearity.

Classical density estimation techniques such as kernel density estimation and mixture of Gaussians, suffer severely from the curse of dimensionality, which limits their model capability for high-dimensional data. However, the approximation of high-dimensional density functions is often needed to alleviate the computational cost of sampling a complicated mathematical model for many engineering applications. For example, a typical Uncertainty Quantification (UQ) model is a partial differential equation (PDE) subject to uncertainty. If we study 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 when sampling the original model \[19, 7, 23\]. Another example is variational Bayes \[2\]. If the forward problem corresponds to solving a PDE, it can be very expensive to compute the statistics with respect to the posterior of a Bayesian inverse problem. Sampling strategies such as MCMC become less effective as the number of dimensions increases. The variational Bayes approach, where the optimal approximation of the distribution will be sought in a family of density models, can often be more effective for high-dimensional problems.

The available deep generative models focus on capturing the main features of the data instead of the accuracy of the estimation 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 employed as a density estimation technique with mathematical convergence. In \[22\], we coupled the real NVP \[5\] 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, say \(O(10)\).

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 distribution of the latent random variables identified by VAE. Using KRnet, we have generalized both the prior and the encoder of the canonical VAE. After doing so, the goal of each component of the model becomes more specific such that VAE-KRnet can be more effective and robust than the canonical VAE. We illustrate this using a linear model between the latent space and the data space. We also apply VAE-KRnet to variational Bayes to approximate the posterior. By varying the number of dimensions of the latent space from zero (KRnet) to \(m\) (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 into the search of latent random variables. We maximize the mutual information between the latent random variable and the original random variable while minimizing 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 varying this parameter, we can obtain the best approximation of the mean and variance. In general the best approximation of the mean and variance cannot be achieved at the same value of the parameter.

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 mapping

Let $\mu_Y$ and $\mu_Z$ indicate the probability measures of random variables $Y$ and $Z$, respectively. In the optimal transport theory, the mapping $T : Z \rightarrow Y$ is called a transport map such that $T_\#\mu_Z = \mu_Y$, where $T_\#\mu_Z$ is the push-forward of the law $\mu_Z$ of $Z$ such that $\mu_Y(B) = \mu_Z(T^{-1}(B))$ for every Borel set $B$ [6]. The Knothe-Rosenblatt rearrangement says that $T$ may have a lower-triangular structure such that

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

It is shown in [3] that such a mapping can be regarded as a limit of a sequence of optimal transport maps when the quadratic cost degenerates. Combining the K-R rearrangement and the technique real-NVP [5], we have proposed an approximation of the invertible mapping $f(\cdot)$ such that $T_\#\mu_Z$ can be used as a model for density estimation when data are provided for $Y$ and a prior distribution is prescribed for $Z$ [21, 22]. We call this model KRnet. In reality, we may consider a block-triangular version of the K-R rearrangement. Consider a partition of $y = (y_1, \ldots, y_K)$, where $y_i = (y_{i,1}, \ldots, y_{i,m})$, where $1 \leq K \leq n$ and $1 \leq m \leq n$, and $\sum_{i=1}^K \dim(y_i) = n$. We then have

$$z = f(y) = \begin{bmatrix}
    f_1(y_1) \\
    f_2(y_1, y_2) \\
    \vdots \\
    f_K(y_1, \ldots, y_K)
\end{bmatrix}. \quad (2.2)$$

Let $\mu_Z(dz) = p_Z(z)dz$, where $p_Z(z)$ is the probability density function (PDF). We have the model for the PDF of $Y$ as

$$p_Y(y) = p_Z(f(y)) |\det \nabla_y f(y)|, \quad (2.3)$$

which can be easily sampled as $Y = f^{-1}(Z)$, thanks to the invertibility of $f(\cdot)$. 
2.1. An overview of the layers in KRnet. The mathematical form of KRnet is an invertible composite function

\[ z = f(y) = f_m \circ f_{m-1} \circ \ldots \circ f_i \circ \ldots f_2 \circ f_1(y), \]  

(2.4)
or

\[ y = f^{-1}(z) = f_m^{-1} \circ f_{m-1}^{-1} \circ \ldots \circ f_i^{-1} \circ \ldots f_2^{-1} \circ f_1^{-1}(z), \]  

(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 \( Y \) step by step through a large number of intermediate simple bijections to make it eventually consistent with a prescribed distribution of \( Z \). We let \( y[0] = y \) indicate the initial state and \( y[i] = f_i \circ \ldots \circ f_1(y) \) an intermediate state. From the viewpoint of dynamical systems, the subscript \( i \) can be regarded as an index for time. The main feature of KRnet is that the overall structure of the invertible mapping is lower (or upper) triangular. More specifically, considering the mapping from \( Y \) to \( Z \), each dimension of \( Y \) will remain fixed at a certain stage until all dimensions become inactive. On the other hand, the inverse mapping from \( Z \) to \( Y \) will activate the dimensions gradually. Before the definition of KRnet, we first briefly introduce all the layers that will be used. Each layer is a relatively simple mapping from the inputs to the outputs.

- **Squeezing layer** deactivates a certain number of components using a mask

\[ q = (1, \ldots, 1, 0, \ldots, 0), \]

which means that the components \( q \odot y[i] \) will keep being updated and the rest components \( (1-q) \odot y[i] \) will remain fixed from then on. Here \( \odot \) indicates the Hadamard product or component-wise product. So we deactivate the last \( n-k \) component by default 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 orthogonal matrix for the current active dimensions:

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

where \( W \in \mathbb{R}^{k \times k} \) with \( k \) being the number of 1’s in \( q \), and \( I \in \mathbb{R}^{(n-k) \times (n-k)} \) is an identity matrix, and \( W = LU \) is the LU decomposition of \( W \). The entries of \( L \) and \( U \) will be treated as trainable parameters of the model except for the diagonal entries of \( L \) which are equal to 1. Intuitively we expect the rotation may put the most important dimensions at the beginning, which need further modifications.

- **Scale and bias layer** provides a simplification of batch normalization which is defined as [11, 13]

\[ \hat{y}[i] = a \odot y[i] + b, \]

(2.6)

where \( a \) and \( b \) are trainable, and initialized by the mean and standard deviation of data. After the initialization, \( a \) and \( b \) 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 \( y_{[i]} = (y_{[i],1}^T, y_{[i],2}^T)^T \) with \( y_{[i],1} \in \mathbb{R}^m \) and \( y_{[i],2} \in \mathbb{R}^{n-m} \). The affine coupling layer is defined as \([22, 5]\)

\[
\begin{align*}
z_1 &= y_{[i],1}, \\
z_2 &= y_{[i],2} \odot (1 + \alpha \tanh(s(y_{[i],1}))) + e^\beta \odot \tanh(t(y_{[i],1})),
\end{align*}
\]

where \( s, t \in \mathbb{R}^{n-m} \) stand for scaling and translation functions depending only on \( y_{[i],1}, 0 < \alpha < 1 \) and \( \beta \in \mathbb{R}^n \). Note that \( y_{[i],2} \) is updated linearly while the mappings \( s(y_{[i],1}) \) and \( t(y_{[i],1}) \) can be arbitrarily complicated, which are modeled as a neural network (NN),

\[
(s, t) = \text{NN}(y_{[i],1})).
\]

Then the Jacobi matrix is lower-triangular, and an inverse can be easily computed. The two parts of \( y_{[i]} \) will be updated interwiningly by a sequence of affine coupling layers, e.g., at the next affine coupling layer, the first partition will be modified while the second partition 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 \( 0 < a < \infty \). For data in \( (-\infty, -a] \cup (a, \infty) \), an identity mapping is considered; for data in \([ -a, a] \), we define

\[
z = \phi^{-1} \circ F \circ \phi(y),
\]

where \( \phi: [-a, a] \to [0, 1] \) is an affine mapping, and

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

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 such that \( F(x) \) is a quadratic function whose inverse can be computed explicitly.

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 and inner loop, where the outer loop has \( K - 1 \) stages, corresponding to the \( K \) mappings \( f_i \) in equation (2.2), and the inner loop has \( L \) stages, indicating the length of an invertible chain of affine coupling layers.

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

\[
z = f(y) = L_N \circ f_{[K-1]}^{\text{outer}} \circ \ldots \circ f_{[1]}^{\text{outer}}(y).
\]

Let \( y_{[k]} = f_{[k]}^{\text{outer}}(y_{[k-1]}) \) with \( y_{[0]} = y \), and \( i = 1, \ldots, K - 1 \). Each \( y_{[k]} = (y_{[k],1}, \ldots, y_{[k],K}) \) has the same partition. The \( i \)th partition will remain unchanged after stage \( K - i + 1 \). For example, \( y_{[k],K} \) will be updated only when \( k = 1 \) and and \( y_{[k],K-1} \) will be fixed when \( k > 2 \). This way, the number of effective dimensions decreases as \( k \) increases. Once the outer loop is completed, the only active dimensions in \( y_{[K-1]} = (y_{[K-1],1}, \ldots, y_{[K-1],K}) \) will be \( y_{[K-1],1} \). We then apply the nonlinear invertible layer to \( y_{[K-1],1} \) before the final output.
\textbf{Figure 1.} Left: the flow chart of the block-triangular invertible mapping.

- **Inner loop.** The inner loop mainly consists of a sequence of general coupling layers \( f_{[k,L]}^{inner} \), which consists of one scale and bias layer and one affine coupling layer, based on which \( f_{[k]}^{outer} \) can be written as:

\[
    f_{[k]}^{outer} = L_S \circ L_N \circ f_{[k,L]}^{inner} \circ \ldots \circ f_{[k,1]}^{inner} \circ L_R,
\]

where \( L_R \) is a rotation layer, and \( L_S \) is a squeezing layer, and \( L_N \) is a nonlinear invertible layer. The nonlinear invertible layer is only applied to the dimensions to be deactivated by the squeezing layer.

**Remark 2.1.** The nonlinear invertible layers and the rotation layers can be switched off to increase the efficiency. The main effectiveness comes from the depth determined by both \( K \) and \( L \).

3. **Coupling VAE and KRnet**

3.1. **Variational autoencoder (VAE).** We now briefly recall the variational autoencoder \([12]\), which provides a dimension reduction technique to estimate the distribution of the data of \( Y \in \mathbb{R}^n \). We assume that there exists a latent random variable \( X \in \mathbb{R}^d \) with \( d \ll n \) with a marginal distribution \( p_{X,\theta}(x) \), where \( \theta \) indicates the model parameters. The joint distribution \( p_{X,Y,\theta}(x,y) \) of \( X \) and \( Y \) is then described by the conditional distribution \( p_{Y|X,\theta}(y|x) \), i.e., \( p_{X,Y,\theta} = p_{X|Y,\theta} p_{Y,\theta} \).

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

$$D_{KL}(q_X|Y^{(i)}, \phi) || p_X|Y^{(i)}, \theta)$$

$$= \int q_X|Y^{(i)}, \phi \log \frac{q_X|Y^{(i)}, \phi}{p_X|Y^{(i)}, \theta} dx$$

$$= \int q_X|Y^{(i)}, \phi \log \frac{q_X|Y^{(i)}, \phi}{p_{Y^{(i)}}(x)} dx$$

$$= D_{KL}(q_X|Y^{(i)}, \phi) || p_X, \theta) - \int q_X|Y^{(i)}, \phi \log p_{Y^{(i)}}X, \theta dx + \log p_{Y^{(i)}} \geq 0. \quad (3.1)$$

The minimization of $D_{KL}(q_X|Y^{(i)}, \phi) || p_X|Y^{(i)}, \theta)$ is the same as the maximization of the variational lower bound of $\log p_Y$, which is defined as 

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

If there exist $N$ samples in the training set, the variational lower bound of $\log p_Y$ is 

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

where $y$ includes all the data $\{y^{(i)}\}_{i=1}^{N}$ in the training set, and $p_Y$ is the likelihood function. Maximizing $\mathcal{L}_{\theta, \phi}(y)$ will yield the optimal choice of $\theta$ and $\phi$.

For application, we need to specify PDF models for $p_{Y^{(i)}|X, \theta}$, $q_{X|Y^{(i)}, \phi}$ and $p_{X, \theta}(x)$. From the viewpoint of dimension reduction, it is often a good choice to assume that $p_{Y^{(i)}|X, \theta}$ is Gaussian with independent components, i.e., $\mathcal{N}(\mu_{de, \theta}(x)), \text{diag}(\sigma_{de, \theta}^2(x)))$, where $\sigma_{de}^2$ means that the square operation is component-wise. The posterior distribution $p_{X|Y^{(i)}, \theta}$ is intractable, and an approximation model $q_{X|Y^{(i)}, \phi}$ is used, which is also chosen as a multivariate Gaussian with independent components, i.e., $\mathcal{N}(\mu_{en, \phi}(y^{(i)})), \text{diag}(\sigma_{en, \phi}^2(y^{(i)})))$. Then $(\mu_{en, \phi}(y), \sigma_{en, \phi}(y))$ serves as the encoder and $(\mu_{de, \phi}(x), \sigma_{de, \phi}(x))$ serves as the decoder. The prior distribution $p_{X, \theta}$ is assumed to be a simple Gaussian $\mathcal{N}(0, I)$. Furthermore, both encoder and decoder are modeled by neural networks.

### 3.2. VAE for a linear model.

Let us consider a linear model for dimension reduction

$$Y = AX + \xi \quad (3.4)$$

where $\xi \in \mathbb{R}^n$, $\sim \mathcal{N}(0, \sigma^2 I)$, $A \in \mathbb{R}^{n \times d}$, and $\xi$ is independent of $X$. We assume that $\sigma$ is small enough such that $X$ can be regarded as a latent random variable with $d < n$. The joint distribution of $X$ and $Y$ is

$$p_{X,Y} = p_X \cdot \mathcal{N}(Ax, \sigma^2 I) \quad (3.5)$$

If we let $p_X = \mathcal{N}(0, I)$ as in the canonical VAE, we have $p_{X|Y} = \mathcal{N}(\mu(y), \Sigma(y))$ with

$$\mu(y) = \sigma^{-2}(I + \sigma^{-2}A^TA)^{-1}A^Ty, \quad \Sigma(y) = (I + \sigma^{-2}A^TA)^{-1}. \quad (3.6)$$

The matrix $\Sigma(y)$ is in general a full matrix except that the column vectors of $A$ are mutually orthogonal.
We now consider a rotation of $X$ as $Z = UX$, where $U$ is a unitary matrix. Letting $p_Z = \mathcal{N}(0, I)$, we have $p_{Y|Z} = \mathcal{N}(AU^T U, \sigma^2 I)$, and
\[
p_{Z|Y} = \mathcal{N}(\mathbf{0}, \mathbf{I} + \sigma^{-2} U A \mathbf{I} A^T U^{-1}). \quad (3.7)
\]

According to the spectral theorem of symmetric matrices, we know there exists a unitary matrix $U$ such that $UA^T AU$ is diagonal. In other words, the encoder $\mathcal{N}(\mu_{en}(y), \sigma_{en}^2(y))$ is able to capture the linear model (3.4) after a rotation of $X$.

If $X$ is a general Gaussian $\mathcal{N}(0, \Sigma_X)$, it can be shown that we need to seek a transformation $Z = UBX$, where $B$ is non-singular and $U$ is unitary, such that the canonical VAE is able to fully capture the linear model. More specifically, the covariance matrix for $Z|y$ is
\[
\Sigma_{Z|y} = (\mathbf{I} + \sigma^2 UB^{-1} A^T A B^{-1} U^T)^{-1}. \quad (3.8)
\]

Letting $Z \sim \mathcal{N}(0, I)$, we have $UB\Sigma_X B^T U^T = \mathbf{I}$, which yields that $B = \Sigma_X^{-1/2}$. It is seen that there exists an unitary matrix $U$ such that $\Sigma_{Z|y} = U\Sigma_X^{-1/2} A^T A \Sigma_X^{-1/2} U^T$ is diagonal.

**Remark 3.1.** For the linear model (3.4) 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 “inverse” of $A$, (2) the mapping $B$, and (3) the rotation $U$. The mapping $B$ transfers a general Gaussian to a standard one, and the rotation $U$ makes the covariance matrix of $Z|y$ diagonal.

### 3.3 Generalize the prior

Assume that $X \sim p_G = \mathcal{N}(0, I)$ following the canonical VAE. We introduce another random variable $Z$ satisfying $X = f_{pr, \beta}(Z)$, where $f_{pr, \beta}()$ is a nonlinear bijection with $\beta$ being the model parameter. We have $p_{Z, \beta}(z) = p_G(f_{pr, \beta}(z))|\nabla_z f_{pr, \beta}(z)|$. We now compare the two cases, where the latent spaces are defined by $X$ and $Z$ respectively. We also assume that $q_{X|Y, \phi}$ and $q_{Z|Y, \phi}$ are defined by the same model, i.e., Gaussian, and so are $p_{Y|X, \theta}$ and $p_{Y|Z, \theta}$. In other words, only the model for the prior is changed. Let
\[
(\phi^*, \theta^*) = \arg \min_{\phi, \theta} \mathcal{L}^X_{\phi, \theta}(y), \quad (3.9)
\]

where the superscript $X$ indicates that the latent space is defined by $X$. For simplicity, we consider $\mathbb{E}_{p_Y} \left[ \mathcal{L}^X_{\phi^*, \theta^*}(Y) \right]$ by noting that
\[
\lim_{N \to \infty} \frac{1}{N} \mathcal{L}^X_{\phi, \theta}(y) = \lim_{N \to \infty} \sum_{i=1}^N \frac{1}{N} \mathcal{L}^X_{\phi, \theta}(y^{(i)}) = \mathbb{E}_{p_Y} \mathcal{L}^X_{\phi, \theta}(Y). \quad (3.10)
\]

More specifically,
\[
\mathbb{E}_{p_Y} \left[ \mathcal{L}^X_{\phi, \theta}(Y) \right] = -D_{KL}(q_{X|Y, \phi} p_Y || p_{X, \theta} p_Y) + \mathbb{E}_{p_Y q_{X|Y, \phi}} \left[ \log p_Y|X, \theta \right]. \quad (3.11)
\]

Note that
\[
\mathbb{E}_{p_Y} \left[ \mathcal{L}^Z_{\phi^*, \theta^*, \beta}(Y) \right] = \mathbb{E}_{p_Y} \left[ \mathcal{L}^X_{\phi^*, \theta^*}(Y) \right]
\]
= $-(D_{KL}(q_{Z|Y, \phi, \beta} p_Y || p_{Z, \beta} p_Y) - D_{KL}(q_{X|Y, \phi, \beta} p_Y || p_{X, \beta} p_Y))$. 

since only the prior depends on $\beta$ and the encoders and decoders are the same for both $X$ and $Z$. It is easy to see that

$$D_{KL}(q_{Z|Y,\phi^*}p_Y||p_{Z,\beta}p_Y) = \int q_{Z|Y,\phi}p_Y \log(p_{Z,\beta})dzdy + \int q_{X|Y,\phi}p_Y \log(p_G)dxdy$$

$$= -\int q_{Z,\phi}p_Y \log(p_{Z,\beta})dz + \int q_{X,\phi}p_Y \log(p_G)dxdy$$

$$= D_{KL}(q_{Z,\phi^*}p_{Z,\beta}) - D_{KL}(q_{X,\phi^*}p_G),$$

where $q_{Z,\phi^*} = q_{X,\phi^*}$ indicates the marginal distribution defined by the joint distribution $q_{Z|Y,\phi^*}p_Y = q_{X|Y,\phi^*}p_Y$. So if $q_{X,\phi^*} \neq p_G$, there always exists $\beta$ such that

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

as long as the model $X = f_{\beta}(Z)$ is good enough, implying that

$$E_{p_Y}[\mathcal{L}^{Z,\phi^*,\beta}(Y)] > E_{p_Y}[\mathcal{L}^{X,\phi^*,\beta}(Y)].$$

In other words, $Z$ provides a better latent space than $X$ since a minimization with respect to $\beta$ will be implemented.

**Remark 3.2.** Since $q_{X|Y,\phi}$ is an approximation of $p_{X|Y}$, equation (3.11) can be understood as

$$-E_{p_Y}[\mathcal{L}^X] \approx I(X, Y) + h(Y|X) = h(Y), \quad (3.12)$$

where the first term on the right-hand side is the mutual information and the second term is the differential conditional entropy. The maximization of $E_{p_Y}[\mathcal{L}^X]$ means that we seek a $X$ that provides most information of $Y$, but is independent of $Y$ as much as possible. So the mutual information $I(X, Y)$ can be regarded as a regularization term. Apparently, if $q_{X|Y,\phi}$ is a good model, we need $p_{X,\theta}$ to be able to describe the real marginal PDF of $X$ such that the latent random variable $X$ can be independent of $Y$ as much as possible. Otherwise, too much regularization might be introduced.

### 3.4. Generalize the encoder.

We now look at the encoder $q_{X|Y}$. Note

$$p_Y p_{X|Y} = p_{X,Y} = p_X p_Y|X,$$

where $p_{X,Y}$ is the joint distribution. In this equation, three PDFs, i.e., $p_Y|X$, $p_{X|Y}$ and $p_X$, will be modeled or approximated by Gaussians in the canonical VAE. Although it is quite straightforward to assume that $p_Y|X$ is Gaussian, e.g., $\mathcal{N}(\mu_{de,\theta}(x), \text{diag}(\sigma_{de,\theta}^2(x)))$, from the viewpoint of model reduction, the choice that $p_{X|Y}$ is also modeled as Gaussian, e.g., $q_{X|Y} = \mathcal{N}(\mu_{en,\phi}(y), \text{diag}(\sigma_{en,\phi}^2(y)))$, is mainly for tractability. Let us consider the following optimization problem

$$\min_{\mu, \Sigma} D_{KL}[p_{X|Y}||\mathcal{N}(\mu, \Sigma)], \quad (3.13)$$
which yields the optimal Gaussian that approximates $p_{X|Y}$. We have
\[
D_{KL}[p_{X|Y} \parallel \mathcal{N}(\mu, \Sigma)] = \int p_{X|Y} \log p_{X|Y} dx \\
+ \int p_{X|Y} \left( \frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) \right) dx \\
+ \int p_{X|Y} \left( \frac{1}{2} \log |\Sigma| + \frac{m}{2} \log(2\pi) \right) dx.
\]
It is seen that
\[
\nabla_\mu D_{KL} = \int p_{X|Y} \Sigma^{-1}(x - \mu) dx = 0,
\]
which yields that
\[
\mu = \int p_{X|Y} x dx = \mathbb{E}_{p_{X|Y}}[X]. \tag{3.14}
\]
For the covariance matrix, we have
\[
\partial_\Sigma D_{KL} = \int p_{X|Y} \left( \frac{1}{2}(x - \mu)(x - \mu)^T - \frac{1}{2} \Sigma \right) dx = 0,
\]
which yields
\[
\Sigma = \int p_{X|Y}(x - \mu)(x - \mu)^T dx = \mathbb{E}_{p_{X|Y}}[(x - \mu)(x - \mu)^T]. \tag{3.15}
\]
Equation (3.15) shows that to approximate $p_{X|Y}$ with a Gaussian, the encoder $\mathcal{N}(\mu_{en,\phi}(y), \text{diag}(\sigma_{en,\phi}^2(y)))$ with a diagonal covariance matrix is in general not enough.

A straightforward way to generalize the encoder $\mathcal{N}(\mu_{en,\phi}(y), \text{diag}(\sigma_{en,\phi}^2(y)))$ is to use $\mathcal{N}(\mu_{en,\phi}(y), \Sigma_{en,\phi}(y))$, where the covariance matrix is defined as a mapping of $y$. We here propose a simpler strategy. We let
\[
f_{en,\alpha}(X|y = \mu_{en,\phi}(y) + \sigma_{en,\phi}(y) \odot f_{en,\alpha}^{-1}(\xi), \tag{3.16}
\]
where $f_{en,\alpha}$ is a flow-based generative model. In other words, we can write
\[
X|y = \mu_{en,\phi}(y) + \sigma_{en,\phi}(y) \odot f_{en,\alpha}^{-1}(\xi), \tag{3.17}
\]
where $\xi \sim \mathcal{N}(0, I)$. If we let $f_{en,\alpha}(\cdot)$ be an identity mapping, the original encoder $\mathcal{N}(\mu_{en,\phi}(y), \text{diag}(\sigma_{en,\phi}^2(y)))$ is recovered. Let us look at the linear model (3.4) again. If the prior $p_X = \mathcal{N}(\mu_{pr}, \Sigma_{pr})$ is an arbitrary Gaussian, it can be obtained that the covariance matrix for $X|y$ is
\[
\Sigma = (\Sigma_X^{-1} + \sigma^{-2} A^T A)^{-1}.
\]
Let $\mu_{en,\phi}(y) = \mathbb{E}_{p_{X|Y}}[X], \sigma_{en,\phi}(y) = 1$ and $f_{en,\alpha}^{-1}(\xi) = \Sigma^{1/2}\xi$. The encoder (3.17) is able to recover $p_{X|Y}$ exactly for the linear model (3.4) with any Gaussian priors. Compared to the canonical VAE, the requirement on the complexity of the encoder is significantly reduced, where the prior can be modeled directly by $f_{pr,\beta}(\cdot)$ and the correlation of $X|y$ is can be taken care of by $f_{en,\alpha}(\cdot)$ (see remark 3.1).
3.5. **VAE-KRnet**. To this end, we have a simple strategy to couple VAE and KR-net. Within the framework of VAE, we keep the original decoder \( \mathcal{N}(\mu_{de,\theta}(x), \text{diag}(\sigma_{de,\theta}^2(x))) \), but generalize the original prior \( p_G \) and the encoder \( \mathcal{N}(\mu_{en,\phi}(y), \text{diag}(\sigma_{en,\phi}^2(y))) \) by incorporating two flow-based generative models \( f_{pr,\beta}(\cdot) \) and \( f_{en,\alpha}(\cdot) \), respectively, as demonstrated in figure 2.

Let us consider the linear model (3.4) subject to a small \( \sigma \) and a non-Gaussian prior \( p_X > 0 \) for any \( x \). Then we have \( p_{X|Y} = p_X p_{Y|X}/p_Y = p_X \mathcal{N}(Ax, \sigma^2 I)/p_Y \). For a smooth function \( g(X) \) of \( X \), we have

\[
E_{p_{X,Y}}[g(X)] = C(y)g(x_0)p_X(x_0)e^{-\frac{|y-Ax_0|^2}{\sigma^2}} + O(\sigma^2),
\]

according to the Laplace’s method, where \( x_0 = (A^TA)^{-1}A^Ty \), which minimizes \( |y-Ax|^2 \) for a given \( y \), and \( C(y) \) is a constant depending on \( y \). Note that the leading term on the right-hand side can also be obtained by integrate \( g(x) \) with respect to the measure \( p_X(x_0)\mathcal{N}(Ax, \sigma^2 I)/p_Y dx \), which corresponds to a multivariate Gaussian distribution subject to a covariance matrix \( (A^TA)^{-1} \), independent of \( y \). This observation means that when \( \sigma \) is small and the underlying relation between \( X \) and \( Y \) is nearly linear, the model (3.17) is effective, which includes \( y \)-dependent scaling and \( y \)-independent correlation, in other words, a component-wise dependence on \( y \) is not necessary for the covariance matrix.

3.6. **Variational lower bound**. We now define the variational lower bound of VAE-KRnet using the following PDFs:

\[
p_{X,\beta} = p_G(f_{pr,\beta}(x)|\nabla x f_{pr,\beta}(x)), \tag{3.19}
\]

\[
q_{X|Y,\phi,\alpha} = p_G \left( f_{en,\alpha} \left( \frac{x - \mu_{en,\phi}(y)}{\sigma_{en,\phi}(y)} \right) \right) |\nabla x f_{en,\alpha}(x)|, \tag{3.20}
\]

\[
p_{Y|X,\theta} = \mathcal{N}(\mu_{de,\theta}(x), \text{diag}(\sigma_{de,\theta}^2(x))). \tag{3.21}
\]

The variational lower bound can be easily approximated by the Monte Carlo method. In particular, the so-called reparameterization trick [4] can be employed. Using equation (3.17), the samples for the PDF \( q_{X|Y,\phi,\alpha} \) can be represented as

\[
x^{(i,k)} = \mu_{en,\phi}(y^{(i)}) + \sigma_{en,\phi}(y^{(i)}) \odot z^{(j)}, \tag{3.22}
\]
where $z^{(j)} = f_{z^{(j)}}^{-1}(\xi^{(i)})$ and $\xi \sim N(0, \mathbf{I})$. The variational lower bound (3.2) will be approximated as

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

(3.23)

where $x^{(i,j)}$ indicates the $k$th sample for $q_{X \mid y^{(i)}, \phi, \alpha}(x \mid y^{(i)})$. For simplicity, we can just let $K = 1$ by noting that

$$
\mathbb{E}_{p_Y} \left[ \mathcal{L}^X \right] = \mathbb{E}_{p_Y q_X \mid Y} \left[ \log \frac{p_{Y \mid X}(y \mid x)}{q_{X \mid y}} \right]
$$

$$
\approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\theta, \phi, \beta, \alpha}^{X}(y^{(i)}, x^{(i)}) = \frac{1}{N} \mathcal{L}_{\theta, \phi, \beta, \alpha}^{X}(y),
$$

(3.24)

where $(y^{(i)}, x^{(i)})$ is corresponds to one sample from the joint PDF $p_{Y \mid X, \phi}$. Then the computation of $\mathcal{L}_{\theta, \phi, \beta, \alpha}^{X}(y)$ based on minibatches can be obtained as

$$
\mathcal{L}_{\theta, \phi, \beta, \alpha}^{X}(y) \approx \frac{M}{M} \sum_{i=1}^{M} \mathcal{L}_{\theta, \phi, \beta, \alpha}^{X}(y^{(i)}, x^{(i)}),
$$

(3.25)

where $M < N$.

4. Density approximation

VAE-KRnet provides a family of probability density models that may be used for density approximation when information about the data or the PDF is available. If $d = n$, i.e., no dimension reduction is considered, VAE-KRnet becomes KRnet. For a fixed number $L$ of general coupling layers $f_{\text{layer}}^j$, the complexity of KRnet relies on $2 \leq K \leq d = n$, i.e., the partition of the dimensions. When $n$ is relatively large, $K = n$ is often not affordable. To reduce the complexity, we have two options: the first option is to let $d = n$ and $K < d$, and the second option is to let $d < n$ and $K \leq d$. The first option corresponds to KRnet with a more aggressive strategy for dimension deactivation, and the second option corresponds to VAE-KRnet with a strategy for dimension reduction. Note that the sample generation is trivial although VAE-KRnet may be quite complex. For simplicity, we use the subscript $*_{\theta}$ to indicate a PDF model with a general model parameter $\theta$ in this section.

4.1. When data are available. So far we have presented VAR-KRnet under the assumption that the data are available. Using the trained PDFs $p_{X, \phi}$ and $p_{Y \mid X, \phi}$, we can approximate marginal PDF of $Y$:

$$
p_{Y, \phi}(y) = \mathbb{E}_{p_{X, \phi}}[p_{Y \mid X, \phi}] \approx \frac{1}{N} \sum_{i=1}^{N} p_{Y \mid X, \phi}(y \mid x^{(i)}),
$$

(4.1)

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

$$
p_{Y, \phi}(y) = \mathbb{E}_{p_{X \mid Y, \phi}} \left[ \frac{p_{Y \mid X, \phi} p_{X, \phi}}{p_{X \mid Y, \phi}} \right] \approx \frac{1}{N} \sum_{i=1}^{N} \frac{p_{Y \mid X, \phi}(y \mid x^{(i)}) p_{X, \phi}(x^{(i)})}{p_{X \mid Y, \phi}(x^{(i)} \mid y)}
$$

(4.2)
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 (4.1).

4.2. **When PDF is available.** For many cases, we need to sample or approximate an arbitrary PDF, e.g., the posterior in Bayesian inference, which is known up to a constant. When the dimensionality is high, all sampling strategies, e.g., MCMC, will suffer from the curse of dimensionality. To alleviate this issue, variational inference is often considered, where the best candidate is sought within a given family of densities by minimizing the KL divergence to the target PDF. Apparently we can apply VAE-KRnet to variational inference.

4.2.1. **KRnet.** Let \( p_Y = C^{-1} \hat{p}_Y \) be a PDF, where \( C \) is an unknown normalization constant, i.e., \( \int \hat{p}_Y dy = C \). Instead of sampling \( p_Y \), we want to sample its approximation \( q_Y, \theta \) given by VAE-KRnet. To train the VAE-KRnet, we in general consider the KL divergence between the PDF model and \( \hat{p}_Y \), where the unknown constant \( C \) will show up as a shift that does not affect the minimization. If dimension reduction is not considered, we employ KRnet and the KL divergence is written as

\[
D_{KL}(q_Y, \theta \| p_Y) = \int q_Y, \theta \log \frac{q_Y, \theta}{p_Y} dy + \log C = D^{pdf}_\theta(\hat{p}_Y) + \log C, \quad (4.3)
\]

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

\[
D^{pdf}_\theta(\hat{p}_Y) = \int q_Z \log \frac{q_Y, \theta(f^{-1}(z))}{\hat{p}_Y(f^{-1}(z))} dz \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{q_Y, \theta(f^{-1}(z^{(i)}))}{\hat{p}_Y(f^{-1}(z^{(i)}))}. \quad (4.4)
\]

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

4.2.2. **VAE-KRnet.** If dimension reduction is considered, the latent random variable \( X \) is taken into account such that the PDF model is given by VAE-KRnet. We minimize the following objective function

\[
- D_{KL}(p_{Y|X, \theta}p_{X, \theta} \| p_{X, \theta}p_{Y}) + \lambda D_{KL}(p_{X, \theta}p_{Y|X, \theta} \| q_{X|Y, \theta}p_{Y}), \quad (4.5)
\]

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

\[
\mathcal{E}^{pdf}_\theta(\hat{p}_Y) = \int p_{Y|X, \theta}p_{X, \theta} \left[ \log \left( \frac{p_{Y|X, \theta}p_{X, \theta}}{\hat{p}_Y} \right) \right]^{\lambda-1} + \log p_{X, \theta} - \log q_{X|Y, \theta}^\lambda dxdy, \quad (4.6)
\]

where \( \lambda > 1 \). When we decrease \( \lambda \) from \( \infty \), the term \( \left( \frac{p_{Y|X, \theta}p_{X, \theta}}{\hat{p}_Y} \right)^{\lambda-1} = 0 \) at \( \lambda = 1 \). Then \( \hat{p}_Y \) 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.

When $\lambda$ goes to infinity, minimizing the objective equation (4.5) is equivalent to minimize directly the KL divergence $DKL(p_Y|x, \theta \| q_X|y, \theta)$. It is seen $p_Y = E_{p_X, \theta}[p_Y|x, \theta]$ as long as $DKL(p_Y|x, \theta \| q_X|y, \theta) = 0$. 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 sufficient accurate. Maximizing the mutual information between $X$ and $Y$ appears to be able to improve the estimation of the variance for properly chosen $\lambda$. In general, we obtain the best prediction of the mean at $\lambda = \infty$, and the best prediction of the variance at a finite $\lambda$. This will be demonstrated later by numerical experiments.

Similar to equation (4.4), we can approximate $E_\theta^{\text{def}}(\hat{p}_Y)$ using the reparameterization trick, where we sample from $N(\theta, I)$ for $(x^{(i)}, y^{(i)}) \in \mathbb{R}^{m+n}$, $i = 1, \ldots, N$. The training process can be implemented in two steps. First, we let $\lambda = \infty$, in other words, we minimize $DKL(p_Y|x, \theta \| q_X|y, \theta)$ only. We can use the trained model to predict the mean. If the variance is needed, we can move on to the second step to continue to train the current model by including the term for mutual information and choosing a finite $\lambda$. We end up with two models, where the first model is for the prediction of the mean and the second model is for the prediction of the variance.

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 $1e^{-3}$ [14]. 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 $f_{pr}(\cdot)$ and $f_{en}(\cdot)$ differs only with respect to the depth or the number of the general coupling layers $f_{\text{inner}}$. 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 $f_{pr}(\cdot)$ (see figure 1), $L_{en}$: the number of general coupling layers in $f_{en}(\cdot)$, and $N_L$: the number of neurons for the neural network in equation (2.9). The training set has $10^5$ samples. Four minimatches are used for the estimation of data distribution, and one minimatch 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 (3.4), where we assume that the column vectors of $A$ are sampled from $N(0, I)$ subject to $\ell_2$ normalization. When $\sigma$ is small, the distribution of $Y$ is mainly a $m$-dimensional distribution of $X \in \mathbb{R}^d$, which is embedded in a $n$-dimensional space. For computation, $Y \in \mathbb{R}^{10}$, and $\sigma = 0.1$. 

For a prescribed prior \( p_X \), we generate the samples of \( Y \) from the linear model (3.4) to form a training set. We will measure the performance of the model using the following quantity (see equation (3.1))

\[
\delta_{\theta,\phi,\beta,\alpha} = \mathbb{E}_{p_Y} \left[ D_{KL}(q_{X|Y,\phi,\alpha}||p_{X|Y,\phi,\alpha}) \right] = -\mathbb{E}_{p_Y} \left[ \mathcal{L}_{\theta,\phi,\beta,\alpha}(Y) \right] - h(Y), \tag{5.1}
\]

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

\[
h(Y) = -\mathbb{E}_{p_Y} \left[ \log p_Y \right] = -\mathbb{E}_{p_Y} \left[ \log \mathbb{E}_{p_X} \left[ p_{Y|X} \right] \right],
\]

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

\[
\mathbb{E}_{p_Y} \left[ \mathcal{L}_{\theta,\phi,\beta,\alpha}(Y) \right] \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\theta,\phi,\beta,\alpha}(y^{(i)}),
\]

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

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

\[
p_X(x) = \mathcal{N}(0, \Sigma_X), \quad p_{Y|X}(y|x) = \mathcal{N}(Ax, \sigma^2 I).
\]

We have

\[
p_Y = \mathcal{N}(0, \sigma^2 I + A \Sigma_X A^T),
\]

which yields that

\[
h(Y) = 5(1 + \log(2\pi)) + \frac{1}{2} \log |\sigma^2 I + A \Sigma_X A^T|.
\]

Let \( \Sigma_X = I \). We sample the two column vectors of \( A \) from \( \mathcal{N}(0, I) \) then normalize them. We know from section 3.2 that the posterior can be recovered by the canonical VAE subject to a rotation of \( X \). 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. 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 \( X \sim \mathcal{N}(0, I) \). To introduce correlation between \( X_1 \) and \( X_2 \), we consider the data satisfying

\[
B = \{ x || Rx^{\alpha,\theta} ||_2 \geq C \},
\]

where \( 0 < \alpha < \infty \), and \( R \) is a matrix defined as

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

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

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

where \( p_{X_i} = \mathcal{N}(0, 1) \) and \( I_B(x) \) 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 $D = 1$, $L_{pr} = L_{en} = 2$, $N_D = 32$, 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 $f_{pr}(\cdot)$ can improve the performance more effectively than $f_{en}(\cdot)$.

We now compare the simulated distributions of $Y$ given by canonical VAE and VAE-KRnet II, where we let $D = 2$, $L_{pr} = 8$, $L_{en} = 2$, $N_D = 32$ 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 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 $X$ provides a latent variable. 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 $X \in \mathbb{R}^3$. For $x = [x_1, x_2, x_3]^T$, we let $x_i = [x_i, x_{i+1}]^T$, $i = 1, 2$, which includes two adjacent components of $x$. We sample from $X \sim \mathcal{N}(0, I)$ and keep the data

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

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 $L_{pr} = 8$, $L_{en} = 2$, $N_L = 24$ 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
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.

has been stuck in a local minimizer until epoch \(\approx 1750\) 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.
5.2. A Bayesian inverse problem. We consider the following linear model for the inverse problem:

\[ \hat{Y} = KY + \xi, \]

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

\[ p_{post}(y) \propto \hat{p}_{post}(y) = \exp \left( -\frac{|\hat{y} - Ky|^2}{2\sigma^2} \right) \exp \left( -\frac{1}{2} |y - \mu_{pr}|^2 \Sigma_{pr}^{-1} \right), \]

what \( \hat{p}_{post}(y) \) is the unnormalized posterior such that \( p_{post}(y) \int \hat{p}_{post}(y)dy = \hat{p}_{post}(y) \), and \( \frac{|y|^2}{\Sigma_{pr}^{-1}} = y^T \Sigma_{pr}^{-1} y \) defines a weighted \( \ell_2 \) norm induced by the precision matrix. We want to find a low-dimension latent random variable \( X \in \mathbb{R}^d \) for \( Y \) such
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 \(L_{pr} = 8\); (d): VAE-KRnet with \(L_{pr} = 10\).

that \(\int p_{X,Y} dx \approx p_{post}(y)\), where \(p_{X,Y}\) is the joint PDF of \(X\) and \(Y\), and will be provided by VAE-KRnet.

We define problem (5.2) using an integral equation

\[
g(x) = \int K(x,y)f(y)dy, \quad (5.4)
\]

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

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

We consider the equation

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

where \(x_i\) are collocation points. This yields a linear system

\[
g = E \Lambda f, \quad (5.7)
\]
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.
where \( e_{ij} = e_i(x_j) \), \( g_j = g(x_j) \), \( i = 1, \ldots, M \), \( j = 1, \ldots, N_x \). We then let \( K = EA \) in equation (5.2) and \( y = g \) 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 \( E \) are nearly mutually orthogonal due to the properties of \( \cos(ix) \).

The condition number of \( K^T K \) depends on the value of \( \gamma \), where the eigenvalue \( \lambda_i^2 \) decays faster for a larger \( \gamma \). We choose \( y_0 = i^{-2.0} \sin(i), \ i = 1, \ldots, n \), and generate the data \( y = Ky_0 + \sigma \xi_0 \), where \( \xi_0 \) is a sample from \( \mathcal{N}(0, I) \). We then consider a Bayesian inverse problem (5.2) using \( y \) as the given data. The true posterior is

\[
p_{\text{post}}(y) = \mathcal{N}(\sigma^{-2}(\Sigma_{pr}^{-1} + \sigma^{-2}K^T K)^{-1}K^T y, (\Sigma_{pr}^{-1} + \sigma^{-2}K^T K)^{-1}).
\] (5.8)

Since we often choose \( \Sigma_{pr} \) as a diagonal matrix, the covariance matrix of \( p_{\text{post}}(y) \) is nearly diagonal by the definition of \( K = EA \), which implies that the components of \( Y \) 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 \( i, j = 1, \ldots, n \), and redefine \( K = EB \). The parameter \( \alpha \) acts as a correlation length. Note that the column vectors of \( EB \) are not nearly mutually orthogonal anymore. Letting \( y = 0 \), the normalization constant for \( p_{\text{post}} \) can be computed as

\[
C = \frac{\hat{p}_{\text{post}}(0)}{p_{\text{post}}(0)} = \exp \left( \frac{1}{2\sigma^2} \| y \|^2 + \frac{1}{2} \| \mu_{pr} \|^2_{\Sigma_{pr}^{-1}} - \frac{1}{2} \| \mu_{\text{post}} \|^2_{\Sigma_{\text{post}}^{-1}} \right),
\] (5.9)

where \( \mu_{\text{post}} \) and \( \Sigma_{\text{post}} \) 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 variables 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 \( \mathcal{N}(\mu_{pr}, \Sigma_{pr}) = \mathcal{N}(0, \Lambda_{pr}) \), where \( \Lambda_{pr} \) is a diagonal matrix with \( \lambda_{pr,i} = i^{-2.5}, i = 1, \ldots, n \). For matrix \( B \), we let \( b_{ij} = e^{-|i-j|/3.0}, i, j = 1, \ldots, n \). Note that from equation (4.4) we have

\[
P_{\theta}^{\text{def}}(\hat{p}_{\gamma}) \geq -\log C.
\]

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 $L_{pr} = 6$, $L_{en} = 2$, $N_L = 24$, $N_D = 32$ for all cases. In $f_{pr}(\cdot)$ and $f_{en}(\cdot)$, the dimensions will be deactivated by two. Let $r(x; Y) = \sum_{i=1}^{n} Y_i e_i(x)$. After we approximate the posterior of $Y$, we compute the $E[r](x)$ 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:

$$\frac{|D_{pdf}^\text{pdf}(\hat{r}_Y) + \log C|}{|\log C|}$$

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 $B$, 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 $\lambda$. After the iteration number reaches $3\times10^5$, 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 seem 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 $\lambda$ on the right. It is seen that the prediction has been improved by increasing $d$, which is as expected. To show the effect of $\lambda$, we compute the errors as follows. Let $\hat{r}(x; Y)$ be an approximation of $r(x; Y)$. We check the following errors:

$$\frac{\|E[\hat{r}] - E[r]\|_{L_2}}{\|E[r](x)\|_{L_2}}$$,

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

for the mean and the standard variation respectively, where the $\| \cdot \|_{L_2}$ is with respect to the space $x$. 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 $\lambda$, VAE-KRnet with a finite $\lambda$ yields a much more accurate estimation of the variance than VAE-KRnet with $\lambda = \infty$. However, VAE-KRnet with a finite $\lambda$ 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 $\alpha$ 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, $-\log C = 632.11$. 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 that case.
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 $\lambda$. Left: varying $d$ with $\lambda = \infty$. Right: varying $\lambda$ with $d = 4$. 
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., $\|E[r]\|_{L_2}$. 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 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.

6. Summary

In this paper, we have developed a family of probability density models by coupling VAE and KRnet. KRnet is an effective invertible mapping without any change in dimension. 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 two things (see Remark 3.1): 1) the “inverse” of the linear system; 2) A mapping from a standard Gaussian to an arbitrary distribution in the latent space. In the canonical VAE, both tasks are achieved mainly by the encoder. In VAE-KRnet, the second task is achieved by KRnet through an invertible mapping. 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) depending on the existence of the latent space. 2) By taking into account the mutual information, a possibility is provided to improve the underestimation of the variance by varying the parameter \( \lambda \), which yields a statistics-oriented way for model selection. 3) Increasing the dimension \( d \) 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 introduce a significant change of the model complexity. 4) For linear Bayesian inverse problems, the KRnet may 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. The current results are very encouraging for us to apply VAE-KRnet to model the posterior of nonlinear Bayesian inverse problems, or to approximate the solution of high-dimensional density equations, which usually have an underlying problem-dependent low-dimensional latent space. The research on these problems will be reported elsewhere.

ACKNOWLEDGMENT

X. Wan has been supported by NSF grants DMS-1620026 and DMS-1913163, and S. Wei has been supported by NSF grant ECCS-1642991.

REFERENCES


Current address, X. Wan: Department of Mathematics, and Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803
E-mail address, X. Wan: xlwan@lsu.edu

Current address, S. Wei: Division of Electrical & Computer Engineering, Louisiana State University, Baton Rouge, LA 70803
E-mail address, S. Wei: swei@lsu.edu