IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Variational Autoencoder with Optimizing Gaussian Mixture Model Priors

Chunsheng Guo[1], Jialuo Zhou[1], Huahua Chen[1], Na Ying[1], Jianwu Zhang[1], Di Zhou[2]

[1] Hangzhou Dianzi University, Hangzhou, 310018 CN

[2]Zhejiang Uniview Technologies Limited Company, Hangzhou, 310051 CN

Corresponding author: Chunsheng Guo (e-mail: guo.chsh@ gmail.com).

ABSTRACT The latent variable prior of the variational autoencoder (VAE) often utilizes a standard Gaussian distribution because of the convenience in calculation, but has an underfitting problem. This paper proposes a variational autoencoder with optimizing Gaussian mixture model priors. This method utilizes a Gaussian mixture model to construct prior distribution, and utilizes the Kullback-Leibler (KL) distance between posterior and prior distribution to implement an iterative optimization of the prior distribution based on the data. The greedy algorithm is used to solve the KL distance for defining the approximate variational lower bound solution of the loss function, and for realizing the VAE with optimizing Gaussian mixture model priors. Compared with the standard VAE method, the proposed method obtains state-of-the-art results on MNIST, Omniglot, and Frey Face datasets, which shows that the VAE with optimizing Gaussian mixture model priors can learn a better model.

INDEX TERMS Variational autoencoder, Gaussian mixture model, Kullback-Leibler distance

I. INTRODUCTION

Variational autoencoder (VAE) [1] recently has become one of the most popular deep generative models. It can capture data distribution through neural networks and be used to generate new samples. Its variants have been widely studied and applied in various fields, such as unsupervised clustering [2] and image generation [3], [4]. VAE [1] belongs to the maximum likelihood generative model. It maximizes Evidence Lower Bound (ELBO) by minimizing model reconstruction errors ("reconstruction loss") and the difference which indicates potential loss through Kullback-Leibler (KL) divergence between the posterior distribution and the hypothesized prior, and then establishes maximum marginal log-likelihood (LL) of the observed data. The deep neural network of the VAE is composed of a bottom-up inference model and a top-down generative model. It is constructed by a function approximator based on backpropagation, which can effectively approximate the intractable posterior distribution. This provides a practical end-to-end generative model that can successfully generate high-quality images [3].

Usually the prior of the VAE utilizes a very simple standard normal distribution, which may over-regularize the posterior distribution, and lead to the problem of posterior collapse, and tend to ignore some latent variable constraints. This often leads to underfitting of the encoder, resulting in a blurred reconstructed image. The commonly used methods to improve the description of latent variables are: 1) defining complex latent variable prior [5], [6], [7]; 2) optimizing and constraining posterior distributions [8], [9], [10], [11], [12], [13], [14]; and 3) constructing a coupling network from posterior to prior [15], [16], [17], [18].

The latent variable of the VAE often utilizes a simple standard normal prior, which may cause the posterior to collapse, making the input signal too weak or too noisy for the posterior parameters, resulting in a limited role of the KL term in ELBO. In specific applications, when the latent variable adopts an inappropriate prior, it may not be able to distinguish between meaningful and meaningless changes in the latent variable [19]. Therefore, many studies have proposed selecting a more suitable latent variable prior. Tomczak and Welling [5] proposed the "Variational Mixture of Posteriors" prior (VampPrior). Chen *et al.* [6] proposed the use of autoregressive flow to learn the prior. Bauer and Mnih [7] proposed a Learned Accept/Reject Sampling (LARS) method, using rejection sampling with a learned acceptance function to construct a richer prior.

Since the KL term in the loss function of VAE is invalidated by posterior collapse, Chou [8] pointed out that adjusting the KL divergence weight can avoid the posterior collapse of the model. Bowman *et al.* [9] proposed that variable weights can be added to the KL divergence in the cost function at training time, as training progresses, this weight increases until it reaches 1, thereby avoiding the posterior collapse of the model. Lin *et al.* [10] believed that the noise level of the currently generative model can be intuitively explained by the prior variance of latent

variables, which can flexibly balance the reconstruction quality and model generalization. Chou and Hathi [11] observed that it is difficult for VAE to map its generated samples to the latent variable space; therefore, during the training process, the effect of samples on latent variables was optimized by introducing a lag. Similarly, Dai and Wipf [12] adopted a two-stage remedy to improve the effectiveness of Gaussian encoder/decoder VAE to generate real samples. Owing to bias issues, good ELBO values do not always imply accurate results. Zhao, Song, and Ermon [13] added a mutual information constraint to help latent variables avoid information preference problems. Rezende and Viola [14] proposed that the VAE of the information bottleneck constraint weighs the reconstruction accuracy and ignores some latent variables (called latent variable collapse), and the combination of optimization and generalization problems leads to other complications.

The invertible mapping or coupling function can adjust the posterior probability density function for adapting to a more complex multimodal probability density, and expand the scalability of the model's posterior approximation capability. Rezende *et al.* [15] proposed a flexible normalized flow (NF) model to approximate the posterior distribution. Through a series of invertible mappings, the model transforms a simple initial probability density function into a more complex multimodal probability density, which greatly expands the scalability of the model's posterior approximation capability. Tomczak *et al.* [16] proposed a method for constructing volume-preserving flows. This method uses a series of Householder transformations with Jacobian determinant equal to 1, which could obtain a lower overhead and maintain the flexibility of the posterior distribution. Ferdowsi *et al.* [17] transformed the VAE's approximate posterior distribution of from the standard Gaussian with diagonal covariance to the first-order autoregressive Gaussian, which provided greater freedom for the approximate posterior to match the true posterior. Cao *et al.* [18] proposed a coupled VAE that used coupled cross-entropy and coupled KL distance to replace the conventional loss function to improve the accuracy and robustness of probability inference on the representation data.

The distance between posterior and prior distribution plays an important role in VAE, forming a regularization term to drive encoder of the VAE to match prior. Besides KL divergence, there are other alternatives for estimating the distance between posterior and prior distributions. Safont *et al.* [19] proposed that the probabilistic distance based on the mixture of non-Gaussian distributions can outperform the KL distance in change detection problems. Tolstikhin *et al.* [20] proposed Wasserstein Auto-Encoders (WAE), which minimized the regularization of the Wasserstein distance between model and target distribution, resulting in a difference form the regularization method used by VAE.

In this paper, we only focus on using complex latent variable priors, and optimizing and constraining the posterior distribution to improve the performance of the variational autoencoder. Recently, a detailed study has been conducted on the effects of the prior and its mismatch with the aggregate variational posterior [22], [23]. It has been shown that the prior that minimizes the KL term in ELBO is given by the corresponding aggregate posterior [5]. However, this choice usually leads to overfitting, because this nonparametric prior will essentially remember the training set. The VampPrior proposed by Tomczak et al. [5] uses the learnable posterior mixture with a fixed number of virtual observations to explicitly model the aggregate posterior. Considering the approximate posterior distribution of VAE as a simple single Gaussian distribution, it is not easy to match a multimodal prior, and the latent variable space may be arbitrarily complex or even multi-peak. To further improve the VAE generation capability, we recommend using a relatively simple Gaussian mixture distribution as the posterior distribution, rather than using a single Gaussian distribution like VampPrior [5] to match the multimodal prior.

In the paper, we propose a new VAE with optimizing Gaussian mixture model priors for improving the model's performance. Our proposed method is called GMMpVAE. Our main contributions of the paper are as follows:

1. A latent variable prior of Gaussian mixture model is proposed, and the aggregated posterior form of the latent variable prior is used to construct the KL distance between posterior and prior distribution of the Gaussian mixture model.
2. Since the KL distance between two mixture Gaussian probability density functions does not have a closed form solution, an approximation of the KL distance is obtained using the greedy algorithm, along with an optimal ELBO approximation.
3. Our proposed method obtains state-of-the-art results on MNIST, Omniglot, and Frey Face datasets.

II. METHOD

A. VARIATIONAL AUTOENCODER

Let $\mathbf{X}=\{\mathbf{x}_1,...,\mathbf{x}_N\}$ is a dataset composed of N independent and identically distributed sample points. Usually, we suppose $\mathbf{x}$ could be generated by some unobserved latent variables $\mathbf{z}$. As $p_\theta(\mathbf{x},\mathbf{z})$ a joint distribution of parametric models, our goal is usually to maximize the average marginal log-likelihood (LL) for the parameters, which can be expressed as $\frac{1}{N}\ln p(\mathbf{X})=\frac{1}{N}\sum_{i=1}^{N}\ln p(\mathbf{x}_i)$. However, because of the intractability of marginal likelihood, when we parameterize the model through a neural network, it leads to the inability to evaluate the marginal likelihood. To overcome this issue, the variational lower bound can be optimized by variational inference:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

$$\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\ln p(\mathbf{x})\right] \geq \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p_\theta(\mathbf{x}|\mathbf{z}) + \ln p_\lambda(\mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})\right]\right] \triangleq \mathcal{L}(\phi,\theta,\lambda) \quad (1)$$

where $q(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x}-\mathbf{x}_n)$ represents empirical distribution, $p_\theta(\mathbf{x}|\mathbf{z})$ represents the generative model (decoder), $p_\phi(\mathbf{z}|\mathbf{x})$ represents the inference model (encoder) and $p_\lambda(\mathbf{z})$ represents the prior, ϕ,θ,λ are their parameters. For continuous latent variables $\mathbf{z}$, the variational lower bound can be effectively optimized by a reparameterization trick [1], [24], the architecture is also known as the variational autoencoder (VAE). During the learning process, we use the L sample points to perform a Monte Carlo estimation of the second term in (1):

$$\tilde{\mathcal{L}}(\phi,\theta,\lambda) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\frac{1}{L}\sum_{l=1}^{L}\ln p_\theta\left(\mathbf{x}|\mathbf{z}_\phi^{(l)}\right) + \ln p_\lambda\left(\mathbf{z}_\phi^{(l)}\right) - \ln q_\phi\left(\mathbf{z}_\phi^{(l)}|\mathbf{x}\right)\right] \quad (2)$$

The first term of (2) represents expectation of negative reconstruction error, which can force the output as close to the input as possible. The second and third terms form a regularization term to drive encoder to match prior.

In general, it is assumed that the encoder has a diagonal covariance matrix, i.e., $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}, \mathrm{diag}\left[\boldsymbol{\sigma}^2\right]\right)$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ represent a D-dimensional vector and are parameterized through a neural network with parameters φ. The prior can be represented by the normal distribution. The decoder can use a suitable distribution according to different data types, for example using the standard normal distribution for continuous data, the Bernoulli distribution for binary data, parameterizing it through a neural network of parameters θ. In addition, non-parametric methods [19], [26] of density estimation can also be used to extend the variational autoencoder framework.

B. THE VARIATIONAL POSTERIOR MIXTURE PRIOR

The training objective (1) is composed of two parts: reconstruction error and regularization term. However, we can also obtain two regularization terms [25] by rewriting (1):

$$\mathcal{L}(\phi,\theta,\lambda) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p_\theta(\mathbf{x}|\mathbf{z})\right]\right] + \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\mathbb{H}\left[q_\phi(\mathbf{z}|\mathbf{x})\right]\right] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\left[-\ln p_\lambda(\mathbf{z})\right] \quad (3)$$

The first term represents the negative reconstruction error, which forces the output as close to the input as possible. The second term represents the expectation of the posterior entropy $H[\cdot]$, which drives encoder to have large entropy for each data point, for example, high variance. The third term represents the cross entropy between aggregated posterior ($q(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n)$) [22], [25] and the prior, which is designed to match the aggregated posterior and the prior.

In general, the prior is preselected like the VAE. However, the prior condition for optimizing ELBO can be found through maximizing the following Lagrange function by Lagrange multipliers β:

$$\max_{p_\lambda(\mathbf{z})} -\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}\left[-\ln p_\lambda(\mathbf{z})\right] + \beta\left(\int p_\lambda(\mathbf{z})\mathrm{d}\mathbf{z} - 1\right) \quad (4)$$

The solution to this problem is to aggregate the posterior:

$$p_\lambda^*(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}|\mathbf{x}_n) \quad (5)$$

However, this choice can cause overfitting [22], [25], and since all training points are calculated each time, the final optimization model will become very expensive. On the other hand, it is well known that a simple prior such as the normal distribution will lead to an over-regularization model, which will have few effective latent dimensions [27].

To solve the problems of high computational complexity, over-regularization and overfitting, the variational posterior mixture generated by pseudo-input can be used to further approximate the optimal solution, i.e., the aggregated posterior:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(\mathbf{z}|\boldsymbol{\alpha}_k) \quad (6)$$

Where K represents the number of pseudo-inputs, $\boldsymbol{\alpha}_k$ is what we call a pseudo-input which is a D-dimensional vector. The pseudo-inputs can be learned by backpropagation, which is considered as a prior hyperparameters, as well as a posterior parameters ϕ, $\lambda = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_K, \varphi\}$. Moreover, the prior produced is multimodal, so it can prevent the regularization of the posterior. On the other hand, once we choose $K \ll N$, combining pseudo-inputs can prevent potential overfitting, which also reduces the training cost of the model. We call this prior the VampPrior.

C. FRAMEWORK BASED ON GAUSSIAN MIXTURE MODEL

The addition of VampPrior improves the ability of generative models. However, the approximate posterior distribution we obtain is still a single Gaussian distribution, which is not easy to match the prior of multimodal, and the latent variable space may be arbitrarily complex or even multi-peak. Obviously, a single Gaussian distribution is not enough to satisfy the distribution of the original latent variable space, and Gaussian mixture density is a very popular representation. Given a sufficient number of components, a Gaussian mixture model [28] can approximate any smoothing function with arbitrary precision. It is a general-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

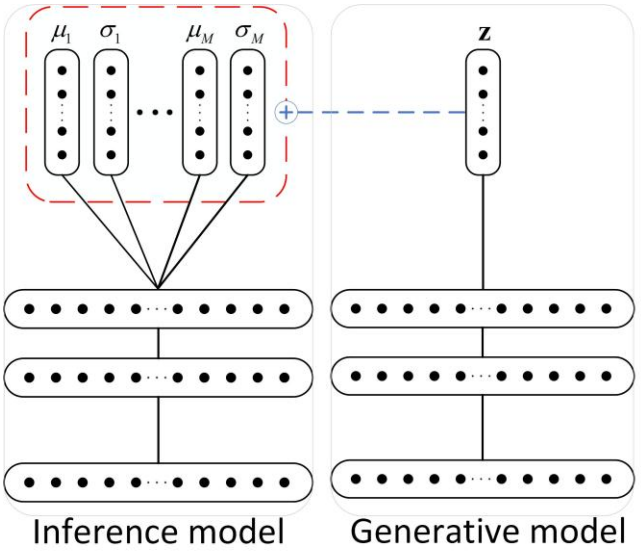

FIGURE 1. GMMpVAE framework. The left solid line bordered rectangle indicates the inference model, and the right solid line bordered rectangle indicates the generative model. As shown by the red dotted bordered rectangle, the posterior distribution of the latent variable is composed of a Gaussian mixture model, while in VAE it is a single Gaussian model. To simplify the calculation, a diagonal covariance matrix is used for each component.

purpose function approximation. Therefore, we propose a framework based on Gaussian mixture model.

There are three items in (2), the first one ($\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[\frac{1}{L}\sum_{l=1}^{L}\ln p_\theta\left(\mathbf{x}\,|\,\mathbf{z}_\phi^{(l)}\right)\right]$) can be regarded as the reconstruction error, and its calculation method is the same as VAE. The second and third terms can be considered as a regularization term, that can be expressed as $-\mathbb{D}_{KL}\left[q_\phi(\mathbf{z}\,|\,\mathbf{x})\,\|\,p_\lambda(\mathbf{z})\right]$. Their mathematical tractability is critical to the overall framework.

Unlike VAE, the regularization term $-\mathbb{D}_{KL}\left[q_\phi(\mathbf{z}\,|\,\mathbf{x})\,\|\,p_\lambda(\mathbf{z})\right]$ presents the KL distance between the two Gaussian mixture densities according to Figure 1, and more details are derived in Section III.

III. KL DISTANCE UPPER BOUND FOR THE DENSITY OF THE TWO MIXTURES

A. KL DISTANCE UPPER BOUND BASED ON RELATIVE ENTROPY CHAIN RULES

The solution of the KL distance between the density of the two Gaussian mixtures ($q_\phi(\mathbf{z}\,|\,\mathbf{x})$ and $p_\lambda(\mathbf{z})$) is crucial in deriving the lower bound in (2), but no closed-form solution can be calculated. Thus, it can only approximate its upper bound [29] by numerical methods. In general, it can approximate its upper bound using the chain rule of relative entropy [29], as shown in Lemma 1.

Lemma 1. The upper bound of the KL distance between the two Gaussian mixture densities $p=\sum_{m=1}^{M}\pi_m f_m$ and $p'=\sum_{m=1}^{M}\pi'_m f'_m$ is:

$$\mathbb{D}_{KL}\left[p\,\|\,p'\right]\le\mathbb{D}_{KL}\left[\pi\,\|\,\pi'\right]+\sum_{m=1}^{M}\pi_m\mathbb{D}_{KL}\left[f_m\,\|\,f'_m\right] \tag{7}$$

Proof: the chain rule of relative entropy. Let X and Y be two random variables, $P(\mathrm{X},\mathrm{Y})$ and $P'(\mathrm{X},\mathrm{Y})$ are two joint distributions, then:

$$\begin{aligned}\mathbb{D}_{KL}\left(P(X,Y)\,\|\,P'(X,Y)\right)=\mathbb{D}_{KL}\left(P(X)\,\|\,P'(X)\right)+\\ \mathbb{D}_{KL}\left(P(Y\,|\,X)\,\|\,P'(Y\,|\,X)\right)\end{aligned} \tag{8}$$

Let S be an unobservable discrete random variable defined in space $\mathcal{S}=\{\mathrm{s}_1,\mathrm{s}_2,...,\mathrm{s}_M\}$, where X is an observable continuous random variable in space R^N, $P(X,S)$ and $P'(X,S)$ are two joint distributions on $\left(\mathrm{R}^N\times\mathcal{S}\right)$, then:

$$\begin{aligned}\mathbb{D}_{KL}\left(P(X,S)\,\|\,P'(X,S)\right)&=\mathbb{D}_{KL}\left(P(X)\,\|\,P'(X)\right)+\\ &\quad\mathbb{D}_{KL}\left(P(S\,|\,X)\,\|\,P'(S\,|\,X)\right)\\ &=\mathbb{D}_{KL}\left(P(S)\,\|\,P'(S)\right)+\\ &\quad\mathbb{D}_{KL}\left(P(X\,|\,S)\,\|\,P'(X\,|\,S)\right)\end{aligned} \tag{9}$$

Because of $\mathbb{D}_{KL}\left(P(S\,|\,X)\,\|\,P'(S\,|\,X)\right)\ge 0$, there are:

$$\begin{aligned}\mathbb{D}_{KL}\left(P(X)\,\|\,P'(X)\right)\le\mathbb{D}_{KL}\left(P(S)\,\|\,P'(S)\right)+\\ \mathbb{D}_{KL}\left(P(X\,|\,S)\,\|\,P'(X\,|\,S)\right)\end{aligned} \tag{10}$$

Let $\pi_m=P(S=s_m)$, $\pi'_m=P'(\mathrm{S}=\mathrm{s}_m)$, $f_m(x)=P(X=x\,|\,S=s_m)$, $f'(x)=P'(X=x)$, and the marginal distribution $P(X=x)=\sum_S P(X=x,S)=\sum_{m=1}^{M}\pi_m f_m(x)$, $P'(X=x)=\sum_S P'(X=x,S)=\sum_{m=1}^{M}\pi'_m f'_m(x)$, let $f(x)=P(X=x)$, $f'(x)=P'(X=x)$, then:

$$\begin{aligned}\mathbb{D}_{KL}\left[p\,\|\,p'\right]&=\mathbb{D}_{KL}\left(P(X)\,\|\,P'(X)\right)\\ &\le\mathbb{D}_{KL}\left[\pi\,\|\,\pi'\right]+\sum_{m=1}^{M}\pi_m\mathbb{D}_{KL}\left[f_m\,\|\,f'_m\right]\end{aligned} \tag{11}$$

If $P(X\,|\,S=s_m)$ and $P'(X\,|\,S=s_m)$ obey the Gaussian distribution, then this upper bound is the KL distance upper bound between two Gaussian mixture models of the same Gaussian number. This upper bound is the same as the upper bound of KL distance derived by log-sum inequality [30].

The KL distance between the two single Gaussians has a closed-form solution [29], so the KL distance upper bound between two Gaussian mixture models can be easily calculated.

B. APPROXIMATE KL DISTANCE UPPER BOUND

It can be seen from (11) that the difference between the joint distribution KL distance and its marginal distribution KL distance causes the difference between the KL distance of the Gaussian mixture model and its upper bound, which shows that the difference in the potential structure between the models is compared in the upper bound of KL distance, that is, the mapping relationship between S and X. Therefore, in order to get a tighter KL distance upper bound, it is necessary to minimize the KL distance between potential mapping relationships within this model. It is worth noting that in the derivation process, there is no mandatory definition of the correspondence between Gaussian components in the two Gaussian mixture models. Therefore, the KL distance upper bound can be minimized by establishing an optimal correspondence between the Gaussian components in the two Gaussian mixture models, thereby better approximating the KL distance.

Let $p(x)=\sum_{m=1}^{M}\pi_m f_m(x)$ and $p'(x)=\sum_{m=1}^{M}\pi'_m f'_m(x)$ be two Gaussian mixture models, where $f_m(\mathrm{x})\sim\mathcal{N}(x;\mu_m,\Sigma_m)$, $f'_m(\mathrm{x})\sim\mathcal{N}(x;\mu'_m,\Sigma'_m)$, define a one-to-one mapping function $\beta: M\to M$, representing the correspondence between Gaussian components in two Gaussian mixture models, then the tighter KL distance upper bound is:

$$\begin{aligned}\mathbb{D}_{KL}(p\,\|\,p')\le\min_{\beta}\sum_{m=1}^{M}\Big\{&\mathbb{D}_{KL}\left[\pi_m\,\|\,\pi'_{\beta(m)}\right]\\&+\pi_m\mathbb{D}_{KL}\left[f_m\,\|\,f'_{\beta(m)}\right]\Big\}\end{aligned}\tag{12}$$

Finding the optimal mapping function $\beta(\cdot)$, minimizing the right end of the above formula requires $M!$ Gaussian distribution operations. Obviously, it is difficult to achieve. We use a greedy algorithm to give a suboptimal mapping function. The specific algorithm is as follows:

1) Sort all Gaussian components in the Gaussian mixture model according to the Gaussian weight $c_1\ge c_2\ge\cdots\ge c_M$ from large to small, such that a, let $i=1$, $A=\varnothing$.

2) Construct the mapping function in turn according to the following formula:

$$\beta(m)=\arg\min_{n\in\{1,\ldots M\}\setminus A}\{D(f_m\,\|\,f'_n)-\log\pi'_n\}\tag{13}$$

Where $A=\{\beta(t)\,|\,t=1,\ldots,m-1\}$.

3) $A=A\cup\{\beta(m)\}$, if $m<M$, $m=m+1$, turn 2); otherwise, end.

Next, start to solve the KL distance upper bound of this model.

As shown in Figure 1, a Gaussian mixture model is proposed to enhance the generative ability of the model. among them:

$$\begin{aligned}q(\mathbf{z}\,|\,\mathbf{x}_i)&=\sum_{m=1}^{M}\pi_m q_m\left(\mathbf{z}\,|\,\boldsymbol{\theta}_m(\mathbf{x}_i)\right)\\&=\sum_{m=1}^{M}\pi_m\mathcal{N}_m\left(\mathbf{z}\,|\,\mu_{i,m},\mathrm{diag}\left[\sigma^2_{i,m}\right]\right)\end{aligned}\tag{14}$$

$$p(\mathbf{z})=\sum_{k=1}^{K}\omega'_k q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)=\sum_{k=1}^{K}\omega'_k\sum_{m=1}^{M}\pi'_m q_m\left(\mathbf{z}\,|\,\boldsymbol{\theta}_m(\boldsymbol{\alpha}_k)\right)\tag{15}$$

Where $q_m\left(\mathbf{z}\,|\,\theta_m(\mathbf{x}_i)\right)=\mathcal{N}_m\left(\mathbf{z}\,|\,\mu_{i,m},diag\left[\sigma^2_{i,m}\right]\right)$, K represents the number of pseudo-inputs, $\boldsymbol{\alpha}_k$ represents pseudo-input, $\omega_k=\omega'_k=1/k$, M represents the number of Gaussian mixture.

$q(\mathbf{z}\,|\,\mathbf{x}_i)$ can be expressed as follows:

$$q(\mathbf{z}\,|\,\mathbf{x}_i)=\sum_{k=1}^{K}\omega_k q(\mathbf{z}\,|\,\mathbf{x}_i)\tag{16}$$

$p_\lambda(\mathbf{z})$ represents the prior of the GMMpVAE, where it is a mixture distribution based on the posterior distribution of the Gaussian mixture model, which can be expressed as follows:

$$p_\lambda(\mathbf{z})=\sum_{k=1}^{K}\omega'_k q_\phi(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)\tag{17}$$

At this point, the KL distance upper bound definition is converted to:

$$\begin{aligned}\mathbb{D}_{KL}\left[q(\mathbf{z}\,|\,\mathbf{x}_i)\,\|\,p(\mathbf{z})\right]&=\mathbb{D}_{KL}\left[\sum_{k=1}^{K}\omega_k q(\mathbf{z}\,|\,\mathbf{x}_i)\,\|\,\sum_{k=1}^{K}\omega'_k q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)\right]\\&=\sum_{k=1}^{K}\omega_k\ln\frac{\omega_k}{\omega'_k}+\\&\quad\sum_{k=1}^{K}\omega_k\mathbb{D}_{KL}\left[q(\mathbf{z}\,|\,\mathbf{x}_i)\,\|\,q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)\right]\end{aligned}\tag{18}$$

Further, $q(\mathbf{z}\,|\,\mathbf{x})$ is a posterior distribution based on the Gaussian mixture model, which can be expressed as follows:

$$q(\mathbf{z}\,|\,\mathbf{x}_i)=\sum_{\mathrm{m}=1}^{M}\pi_m\mathcal{N}_m\left(\mathbf{z}\,|\,\mu_{i,m},\mathrm{diag}\left[\sigma^2_{i,m}\right]\right)\tag{19}$$

Similarly, $q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)$ can be expressed as follows:

$$q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)=\sum_{m=1}^{M}\pi'_m\mathcal{N}_m\left(\mathbf{z}\,|\,\mu_{k,m},\mathrm{diag}\left[\sigma^2_{k,m}\right]\right)\tag{20}$$

At this point, the mapping function $\beta(\cdot)$ is solved by combining the greedy algorithm proposed above, and $\mathbb{D}_{KL}\left[q(\mathbf{z}\,|\,\mathbf{x}_i)\,\|\,q(\mathbf{z}\,|\,\boldsymbol{\alpha}_k)\right]$ can be further expressed as:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

$$\begin{aligned}\mathbb{D}_{KL}\left[q(\mathbf{z}\mid\mathbf{x}_i)\,\|\,q(\mathbf{z}\mid\boldsymbol{\alpha}_k)\right]&=\sum_{m=1}^{M}\mathbb{D}_{KL}\left(\pi_m\,\|\,\pi'_{\beta(m)}\right)+\\ \pi_m\mathbb{D}_{KL}&\left[\mathcal{N}_m\left(\mathbf{z}\mid\mu_{i,m},\Sigma_{i,m}\right)\|\,\mathcal{N}_{\beta(m)}\left(\mathbf{z}\mid\mu'_{k,\beta(m)},\Sigma'_{k,\beta(m)}\right)\right]\\ &=\sum_{m=1}^{M}\pi_m\ln\frac{\pi_m}{\pi'_{\beta(m)}}-\frac{D}{2}+\\ \frac{1}{2}\sum_{m=1}^{M}\pi_m&\left[\log\frac{|\Sigma'_{k,\beta(m)}|}{|\Sigma_{i,m}|}+Tr\left[\Sigma'^{-1}_{k,\beta(m)}\Sigma_{i,m}\right]+\right.\\ &\left.\left(\mu_{i,m}-\mu_{k,\beta(m)}\right)^T\Sigma'^{-1}_{k,\beta(m)}\left(\mu_{i,m}-\mu_{k,\beta(m)}\right)\right]\end{aligned}\tag{21}$$

In summary, the upper bound of KL distance between two Gaussian mixture densities is defined as follows:

$$\begin{aligned}\mathcal{U}_{i,KL}\overset{\text{def}}{=}&\sum_{k=1}^{K}\omega_k\ln\frac{\omega_k}{\omega'_k}+\sum_{k=1}^{K}\omega_k\left[\sum_{m=1}^{M}\pi_m\ln\frac{\pi_m}{\pi'_{\beta(\mathrm{m})}}-\frac{D}{2}+\right.\\ &\frac{1}{2}\sum_{m=1}^{M}\pi_m\left[\log\frac{|\Sigma'_{k,\beta(\mathrm{m})}|}{|\Sigma_{i,m}|}+Tr\left[\Sigma'^{-1}_{k,\beta(\mathrm{m})}\Sigma_{i,m}\right]+\right.\\ &\left.\left.\left(\mu_{i,m}-\mu_{k,\beta(\mathrm{m})}\right)^T\Sigma'^{-1}_{k,\beta(\mathrm{m})}\left(\mu_{i,m}-\mu_{k,\beta(\mathrm{m})}\right)\right]\right]\end{aligned}\tag{22}$$

IV. ALGORITHM SUMMARY

Algorithm 1 Variational Inference with GMMpVAE

Input: dataset of images $\mathcal{D}_{train}=\{\mathrm{x}_n\}_1^N$

Output: train model parameters ϕ,θ (inference/ generative model) and λ

Initialize parameters of inference model ϕ and generative model θ by he initialization

Initialize parameters of the GMMpVAE prior λ

While not converged do

$\mathbf{x}\leftarrow$ {Get mini-batch from dataset}

$\mathbf{z}\leftarrow$ Generated from GMM $q_\phi(\mathbf{z}\mid\mathbf{x})$ using (16)

$\mathcal{L}_{\text{GMMpVAE}}\leftarrow$ Obtain the lower bound using (23)

$\Delta\theta\leftarrow\nabla_\theta\mathcal{L}_{\text{G}}$

$\Delta\phi\leftarrow\nabla_\phi\mathcal{L}_{\text{G}}$

$\Delta\lambda\leftarrow\nabla_\lambda\mathcal{L}_{\text{G}}$

$\theta,\phi,\lambda\leftarrow$ Update parameters using gradients $\Delta\theta$, $\Delta\phi$ and $\Delta\lambda$

end while

According to $\mathcal{U}_{i,KL}\geq\mathbb{D}_{KL}\left[q_\phi(\mathbf{z}\mid\mathbf{x}_i)\,\|\,p_\lambda(\mathbf{z})\right]$ and (22), the variational lower bound (1) can be redefined as:

$$\ln p(\mathbf{x})\geq\mathcal{L}\geq\mathcal{L}_{\text{GMMpVAE}}\overset{def}{=}\mathbb{E}_{q(\mathbf{z}\mid\mathbf{x})}\left[\ln p_\theta(\mathbf{x}\mid\mathbf{z})\right]-\mathcal{U}_{i,KL}\tag{23}$$

The summary is shown in Algorithm 1.Based on the modification of VAE [1], this algorithm can obtain the derivatives $\Delta\theta,\Delta\varphi$ and $\Delta\lambda$, and can use the obtained gradient with stochastic optimization methods (e.g., SGD and ADAM [31]). The main differences from VAE are: first, a new prior is constructed by the aggregated posterior form of the latent variable prior and a Gaussian mixture model; second, the new lower bound is solved.

V. EXPERIMENT

In this section, we introduce the data sets and related experimental settings used in the experiments in section V-A. Then in section V-B, the influence of network structures, the number of mixture coefficients, and the number of pseudo-inputs on the lower bound of the marginal LL is analyzed. As detailed in section V-C, our proposed method obtains state-of-the-art results on the MNIST, Omniglot, and Frey Faces datasets, compared to the generative models based on VAEs. Next, we analyze the reconstruction results of different number of components in latent spaces of different dimensions in section V-D. Finally, section V-E gives a histogram of the LL of the test set to help us analyze how the flexibility of the prior distribution affects them. According to the experimental results, the VAE with optimizing Gaussian mixture model priors can learn a better model.

A. DATASET AND EXPERIMENTAL SETUP

We experimented with three image datasets: dynamic MNIST, Omniglot, and Frey Faces.

1) DATASET

MNIST: We evaluate the model on the standard MNIST dataset (handwritten image dataset) [32] in this experiment. It contains 60,000 training images which we divide into 50,000 training images and 10,000 validation images to adjust the hyperparameters for model selection and 10,000 test images. For model selection, we divided 60,000 training images into 50,000 training images and 10,000 validation images for adjusting the hyperparameters. However, there are two binarization methods for generative models in the literature, which produce different lower bound values. Here we select dynamically binarized MNIST for all experiments [33].

Omniglot: Omniglot [34] is an encyclopedia of writing systems and languages for learning from a small number of examples. Essentially, these characters are similar to MNIST, containing 1,623 handwritten characters from 50 different letters, each of which is represented by approximately 20 images. Here, we divide the data set into 24,345 training data points and 8070 test images, and randomly select 1345 samples from the training set as the validation set. We use dynamically binarized data during training, as with the dynamically binarized MNIST.

Frey Faces: Frey Faces is a face dataset with different emotions taken from consecutive frames of video. Here, the number of training images, validation images, and test

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

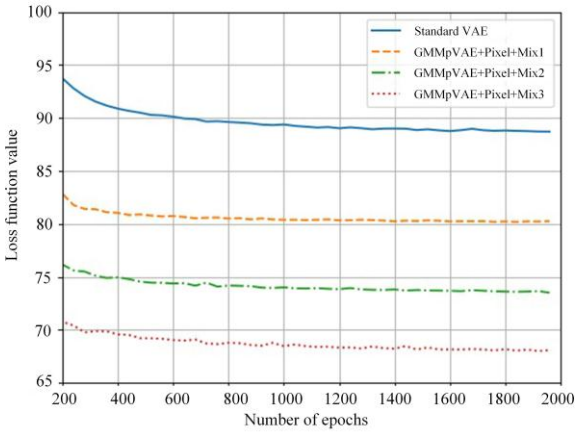

FIGURE 2. Comparison of loss function value of our method using VAE and different M values on the MNIST dataset.

TABLE 1. Marginal log-likelihood (LL) of MNIST test set under different experimental parameter choices.

Method	$\leq \ln p(\mathbf{x})$	RE	KL
VAE[1]	-93.89 $\pm$ 0.09	65.80	28.10
VAE+VampPrior (k=500) [5]	-82.38		
GMMpVAE+MLPs (k=100)	-85.90 $\pm$ 0.19	61.28	24.62
GMMpVAE+MLPs (k=500)	-85.32 $\pm$ 0.12	60.29	25.03
GMMpVAE+MLPs (k=1000)	-85.53 $\pm$ 0.24	60.19	25.34
GMMpVAE+CNN (k=100)	-83.82 $\pm$ 0.14	58.27	25.55
GMMpVAE+CNN (k=500)	-83.21 $\pm$ 0.08	57.23	25.98
GMMpVAE+CNN (k=1000)	-83.43 $\pm$ 0.18	57.44	25.99
GMMpVAE+Pixel (k=100)	-79.96 $\pm$ 0.11	67.86	12.1
GMMpVAE+Pixel (k=500)	-79.66 $\pm$ 0.15	66.47	13.19
GMMpVAE+Pixel (k=1000)	-79.75 $\pm$ 0.16	66.21	13.54
GMMpVAE+MLPs+Mix2 (k=100)	-78.63 $\pm$ 0.27	53.77	24.86
GMMpVAE+MLPs+Mix2 (k=500)	-78.24 $\pm$ 0.23	53.26	24.98
GMMpVAE+MLPs+Mix2 (k=1000)	-78.57 $\pm$ 0.28	53.56	25.01
GMMpVAE+CNN+Mix2 (k=100)	-76.15 $\pm$ 0.21	50.04	26.1
GMMpVAE+CNN+Mix2 (k=500)	-75.84 $\pm$ 0.16	49.65	26.19
GMMpVAE+CNN+Mix2 (k=1000)	-76.03 $\pm$ 0.19	49.82	26.21
GMMpVAE+Pixel+Mix2 (k=100)	-73.08 $\pm$ 0.11	47.64	25.44
GMMpVAE+Pixel+Mix2 (k=500)	-72.72 $\pm$ 0.09	47.13	25.59
GMMpVAE+Pixel+Mix2 (k=1000)	-72.91 $\pm$ 0.18	47.27	25.64
GMMpVAE+MLPs+Mix3 (k=100)	-74.69 $\pm$ 0.33	51.43	23.25
GMMpVAE+MLPs+Mix3 (k=500)	-73.83 $\pm$ 0.24	50.51	23.33
GMMpVAE+MLPs+Mix3 (k=1000)	-74.00 $\pm$ 0.21	50.61	23.39
GMMpVAE+CNN+Mix3 (k=100)	-71.10 $\pm$ 0.17	45.74	25.36
GMMpVAE+CNN+Mix3 (k=500)	-70.61 $\pm$ 0.13	45.15	25.46
GMMpVAE+CNN+Mix3 (k=1000)	-70.79 $\pm$ 0.16	45.28	25.51
GMMpVAE+Pixel+Mix3 (k=100)	-67.88 $\pm$ 0.12	41.31	26.57
GMMpVAE+Pixel+Mix3 (k=500)	-67.46 $\pm$ 0.10	40.83	26.63
GMMpVAE+Pixel+Mix3 (k=1000)	-67.58 $\pm$ 0.19	40.93	26.65

images are randomly divided into 1565, 200, and 200, respectively. The input to this dataset is in grayscale format.

2) EXPERIMENTAL SETUP

We first modeled the distribution using multilayer perceptrons (MLPs) with permutation-invariant settings. In the experiment, MLPs were two hidden layers with three hundred hidden units. Next, we chose the gating mechanism [35] as elementwise nonlinearity. For the latent variable z, we chose 40 random hidden units. Then, we utilized a convolutional neural network (CNN) with gating mechanism instead of MLPs. Finally, we utilized pixelCNN [36] as the decoder. Since the MNIST and Omniglot datasets are binary data, we utilized the Bernoulli distribution; since the Frey Faces dataset is the grayscale format, we utilized the logistic distribution.

We used He initialization [37] to initialize the weight and used ADAM [31] with the mini-batch size equal to 100. In addition, in order to improve the decoder's generative ability, we used the warm-up [9] for 200 epochs. The maximum epochs were 5000. The early-stopping with a look ahead of 100 epochs was used.

We formulated the following rules based on different parameters in each experiment: GMMpVAE + network architecture + number of components of the Gaussian mixture model + number of pseudo-inputs. For example, in GMMpVAE+CNN+Mix3 (k=500), CNN represents the experimental network architecture, Mix3 represents the three component Gaussian mixture model, and k is the number of pseudo-inputs used in this experiment. In the network architecture, MLPs are represented by a multi-layer perceptron, CNN means both the encoder and the decoder are convolutional layers, Pixel means that the encoder is a convolutional layer, and the decoder is PixelCNN [36]. The above network structures are all used gating mechanism [35].

B. COMPARISON OF DIFFERENT NETWORK STRUCTURES, THE NUMBER OF MIXTURE COEFFICIENTS, AND THE NUMBER OF PSEUDO-INPUTS

We tried different choices of parameters on the MNIST, Omniglot, and Frey Faces datasets to analyze the effects of different network structures, the number of mixture coefficients and the number of pseudo-inputs. Figure 2 shows the loss function of the MNIST dataset with different numbers of mixture coefficients, that is, the lower bound of the negative LL. Tables 1 and 3 collect different combinations of parameters for MNIST and Omniglot. Tables 2, 3 and 4 collect the results of our method and the latest method for MNIST, Omniglot, and Frey Faces, respectively. The lower bound of the marginal LL of the numbers is represented by $\leq p(\mathbf{x})$ in the table, the reconstruction error is represented by RE, and the Kullback-Leibler distance is represented by KL. To calculate the mean and deviation, we repeated each of the above experiments thrice.

In Figure 2, we showed the loss function comparison of our method using VAE and different M values on the MNIST dataset. Since we used warm-up in the first 200 epochs, we plotted the loss function as 200 to 2000 epochs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

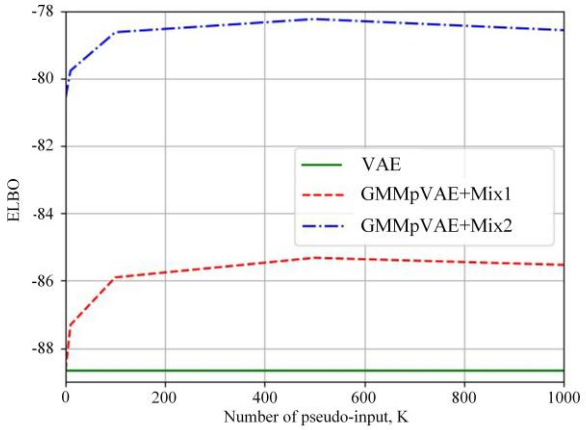

FIGURE 3. In ELBO, the comparison between VAE, GMMpVAE+Mix1 and GMMpVAE+Mix2 and the effect of the number of pseudo-inputs on dynamically binarized MNIST.

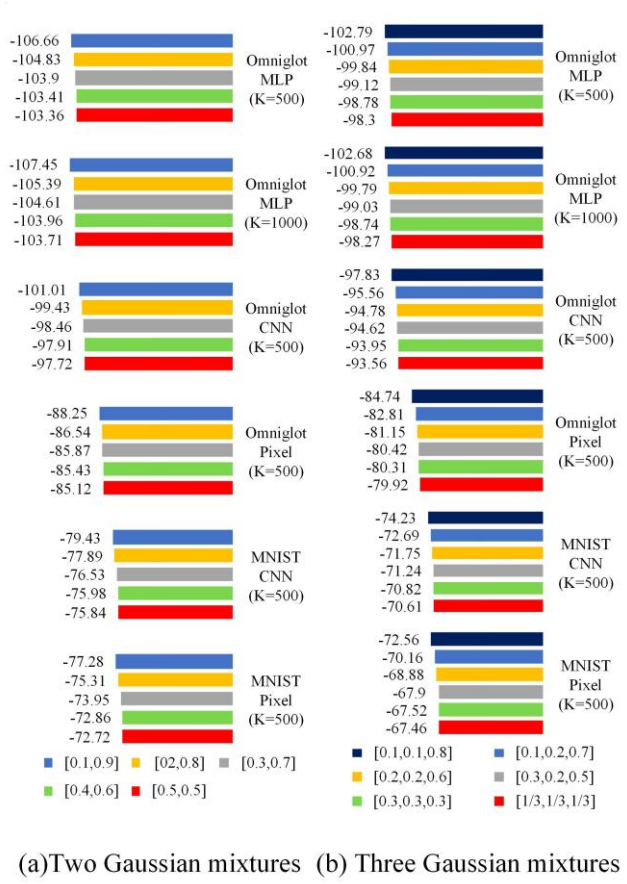

FIGURE 4. Comparison of marginal LL lower bound under different mixture coefficients.

The curves in the figure correspond to VAE and our method from top to bottom, respectively, with three different M, 1, 2, and 3. It can be seen from Figure 2 that our results were much better than VAE and M = 1 even when M = 2 or M = 3 on the MNIST dataset.

TABLE 2. Marginal LL lower bounds of MNIST test set.

Method	$\leq \ln p(\mathbf{x})$
VAE [1]	-93.89 $\pm$ 0.09
VAE+NF(T=80) [15]	-85.10
VAE+NICE(T=80) [38]	-87.20
Conv.VAE+HVI [39]	-81.94
DBN [42], [43]	-84.55
AVAE [44]	-82.97
IWAE [27]	-82.90
Ladder VAE [45]	-81.74
DRAW+VGP [40]	-79.88
VAE+IAF [41]	-79.10
VLAE [6]	-78.53
PixelHVAE(L=2)+VampPrior(k=500) [5]	-78.45
ConvHVAE(L=2)+LarsPrior [46]	-80.30
GMMpVAE+CNN+Mix3 (k=500)	-70.61 $\pm$ 0.13
GMMpVAE+Pixel+Mix3 (k=500)	**-67.46 $\pm$ 0.10**

TABLE 3. Marginal LL lower bounds of Omniglot test set.

Method	$\leq \ln p(\mathbf{x})$
VAE [1]	-106.31
IWAE [27]	-103.38
RBM [47]	-100.46
Conv DRAW [3]	>-91.00
VLAE(fine-tuned) [6]	-89.83
DBN [42], [43]	-100.45
Discrete VAE [42]	-97.43
PixelHVAE(L=2)+VampPrior (k=500) [5]	-89.76
ConvHVAE(L=2)+LarsPrior [46]	-97.08
GMMpVAE+CNN+Mix3 (k=500)	-93.56 $\pm$ 0.15
GMMpVAE+Pixel+Mix3 (k=500)	**-79.92 $\pm$ 0.09**

TABLE 4. Marginal LL lower bounds of Frey Face test set.

Method	$\leq \ln p(\mathbf{x})$
VAE [1]	-1831.11
VAE+NF(T=80) [15]	-1798.93
VAE+NICE(T=80) [38]	-1807.68
IWAE [27]	-1782.73
VAE+IAF [41]	-1759.46
ConvHVAE(L=2)+VampPrior (k=500) [5]	-1782.33
PixelHVAE(L=2)+VampPrior (k=500) [5]	-1742.82
GMMpVAE+Pixel+Mix3 (k=500)	**-1685.62 $\pm$ 17.52**

Table 1 shows the results divided into 4 groups separated by horizontal lines. The basic network structure is in the first group. In the second, third and fourth groups, we selected experimental results for different network structures, the number of mixture coefficients and the number of pseudo-inputs. Among them, using different network structures and the number of pseudo inputs, the results of the second group were actually equal to VAE+Vampprior [5], because the results were obtained using a single Gaussian model. The last two sets show the results obtained using different network structures and the number of pseudo-inputs when the number of mixture coefficients was 2 and 3, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

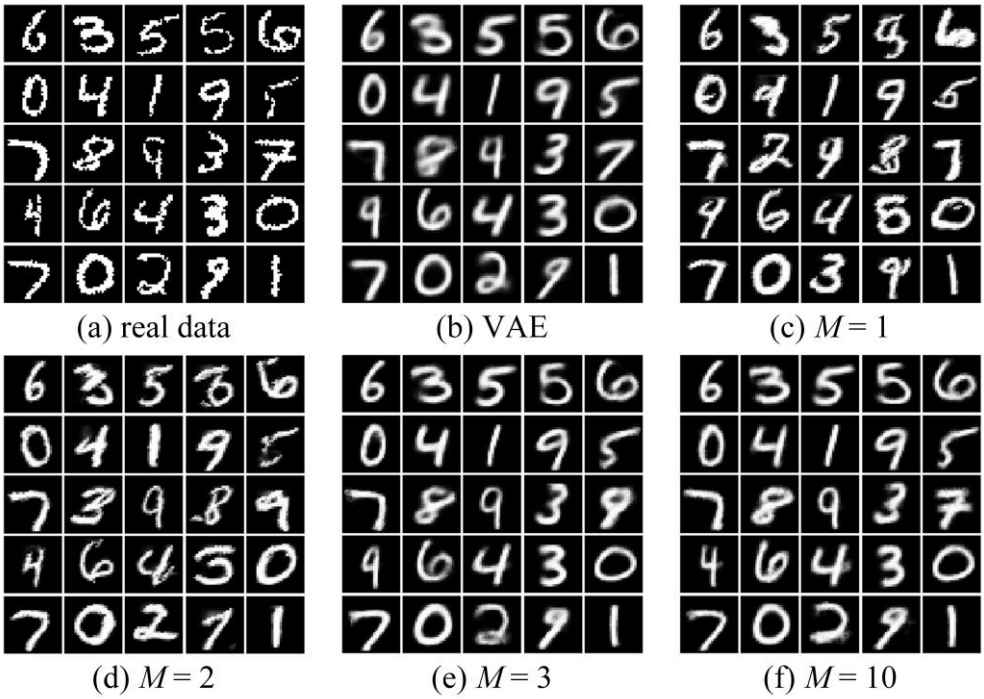

FIGURE 5. On the MNIST dataset, the reconstruction results using VAE and GMMpVAE with different M values when the latent variable dimension is 2.

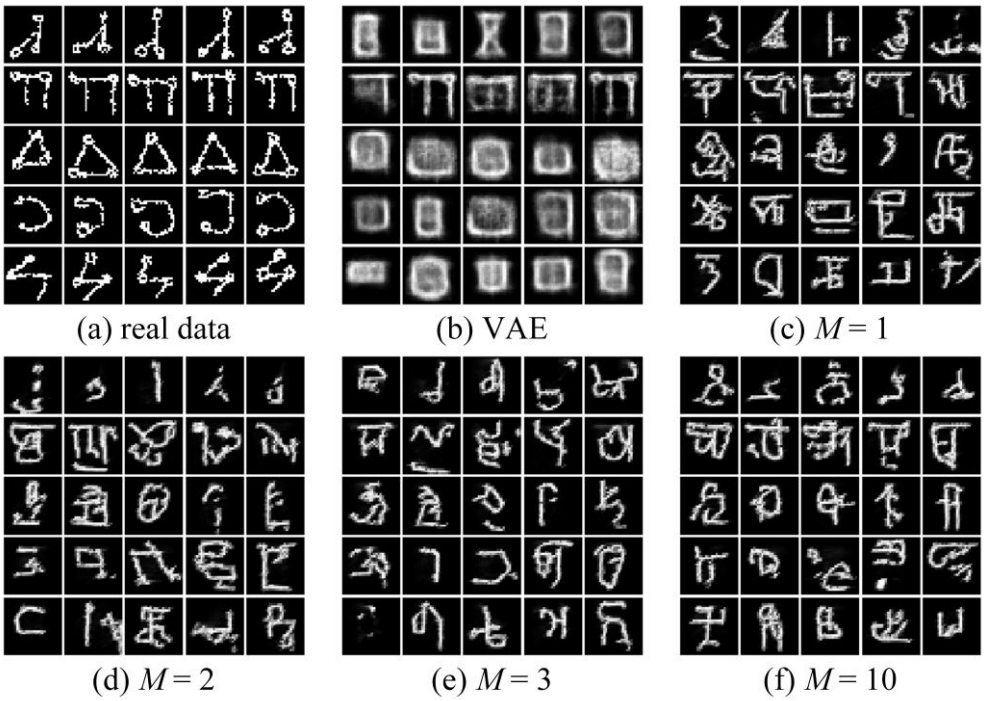

FIGURE 6. On the Omniglot dataset, the reconstruction results using VAE and GMMpVAE with different M values when the latent variable dimension is 2.

First, the performance gradually became better as the number of mixture coefficients increased under the same network structure. At the same time, the reduction of KL and RE represented smaller penalty and lower reconstruction error, respectively, which indicated the effectiveness of our method in improving the generative performance. Second, we noted that as shown in Figure 3 the new prior in this method significantly improved the generative performance than the standard normal prior in all cases, indicating that the combination of multimodal prior and mixture posterior is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

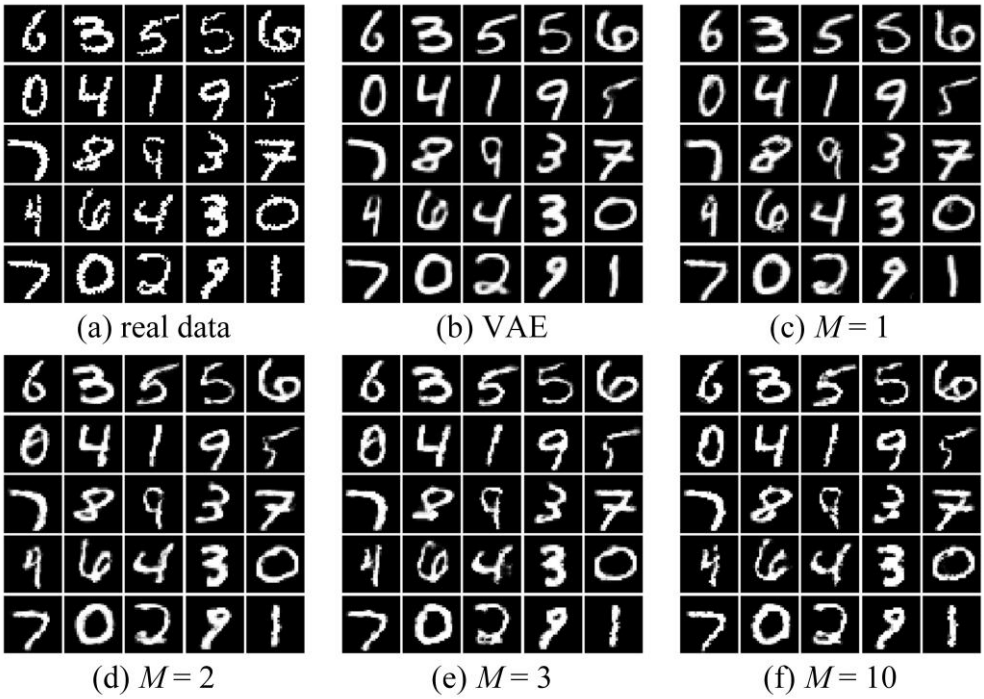

FIGURE 7. On the MNIST dataset, the reconstruction results using VAE and GMMpVAE with different M values when the latent variable dimension is 40.

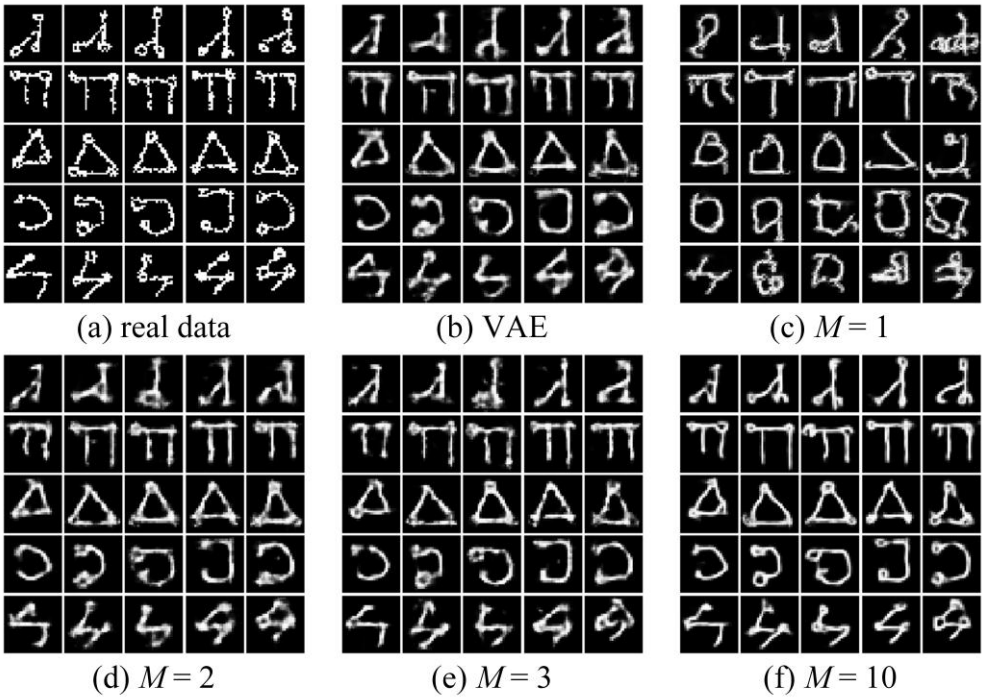

FIGURE 8. On the Omniglot dataset, the reconstruction results using VAE and GMMpVAE with different M values when the latent variable dimension is 40.

better than the standard normal prior. Surprisingly, Figure 3 shows us that the use of more pseudo-inputs did not contribute to improving generative performance, rather, its use resulted in a decrease in generative performance. Through experiments, we found that the model had the best performance when using 500 pseudo inputs for the MNIST and Omniglot datasets.

In order to compare the effects of different mixture coefficients on the marginal log-likelihood lower bound, we designed two sets of experiments to compare the effects of the marginal LL lower bound on different mixture coefficients. The first group used two Gaussian mixtures, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

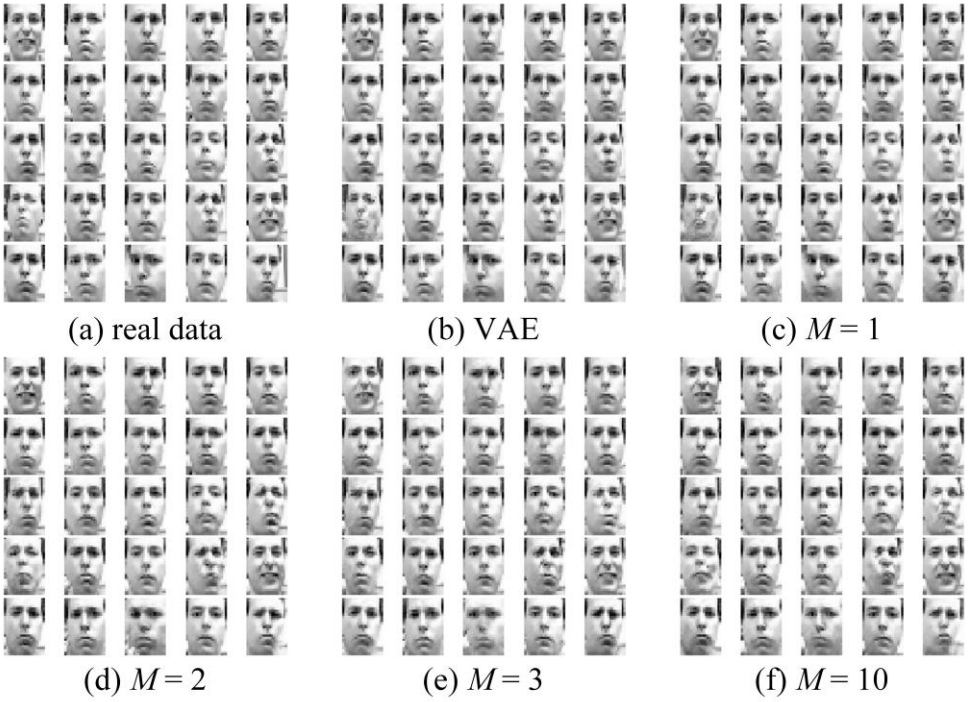

FIGURE 9. On the Frey Face dataset, the reconstruction results using VAE and GMMpVAE with different M values when the latent variable dimension is 40.

the mixture coefficient values were set to [0.1, 0.9], [0.2, 0.8], [0.3, 0.7], [0.4, 0.6], and [0.5, 0.5], respectively. The second group used three Gaussian mixtures, and the mixture coefficient values were set to [0.1, 0.1, 0.8], [0.1, 0.2, 0.7], [0.2, 0.2, 0.6], [0.3, 0.2, 0.5], [0.3, 0.3, 0.4] and [1/3, 1/3, 1/3], respectively. As shown in Figure 4, for detailed comparisons, we used different datasets, the number of network structures and pseudo-inputs tested six experiments on each group. Different colors correspondingly represented the marginal LL lower bound obtained by different mixture coefficients in each experiment.

It can be seen that different mixture coefficients have only a small effect on the results, so in the following experiments, we chose a fixed mixture coefficient. In other words, we set the corresponding coefficient $\pi_M = 1/M$ to the M components of the Gaussian mixture model. For the M components of the Gaussian mixture model, the corresponding coefficient was set to $\pi_M = 1/M$.

C. COMPARISON OF MNIST, OMNIGLOT, and FREY FACES DATASETS WITH TRADITIONAL METHODS BASED ON VAE

Our proposed method obtained state-of-the-art results on the MNIST, Omniglot, and Frey Faces datasets, compared to the generative models based on VAEs.

We selected several effective unsupervised generative models based on VAEs for comparison with our proposed method as shown in Table 2, which utilized different techniques for improving the model's generative ability. For example, inverse Autoregressive Flow (IAF) [41], variational lossy autoencoder (VLAE) [6], variational Gaussian processes (VGP) [40], importance weighted autoencoders (IWAE) [27], Hamiltonian variational inference (VAE+HVI) [39], linear normalized flow (VAE+NF) [15], auxiliary variational autoencoder (AVAE) [44], and so on. Our proposed method obtained state-of-the-art result of these benchmarks.

In the Omniglot datasets, the shape changes of different letters were more complicated than other datasets. When the network structure was CNN, the number of components of the Gaussian mixture model must be increased for improving its generative ability, but when we used pixelCNN as the decoder, only the three Gaussian mixture components obtained state-of-the-art results. GMMpVAE was significantly better than other methods as shown in Table 3.

The results of the Frey Faces dataset were divided into two groups by a horizontal line as shown in Table 4. The results of the first group were obtained from VAE, IWAE, PixelHVAE (L=2) + VampPrior (k=500), and so on, all of which applied the gating mechanism. Therefore, our method also used the same gating mechanism for more efficient comparisons. Our method was better than the compared methods, including VAE, PixelHVAE (L = 2) + VampPrior (k = 500), and so on under the same gating mechanism. At the minimum it has been proven that using the Gaussian mixture model can improve the model's generative ability effectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

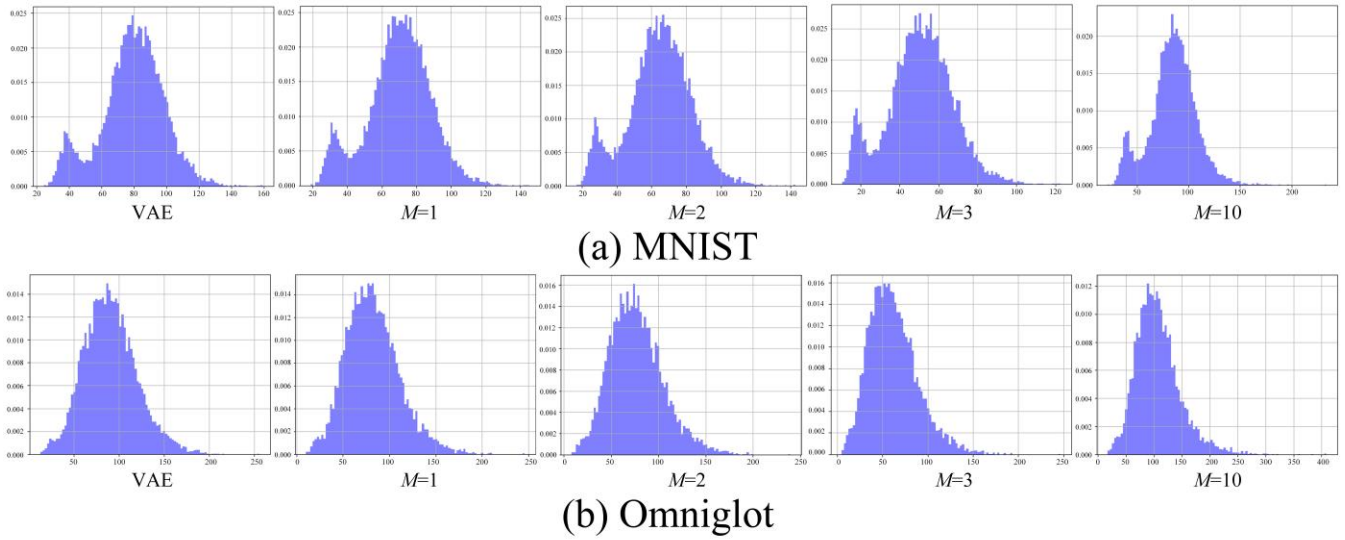

FIGURE 10. Test set LL histogram of the MNIST and Omniglot datasets. The results of each row represented VAE and our proposed method with the different number of components M. The abscissa of each histogram represented the LL value, and the ordinate represented the probability.

D. RECONSTRUCTION RESULTS OF DIFFERENT COMPONENT QUANTITIES IN GAUSSIAN MIXTURE MODEL

Choosing the number of components in the Gaussian mixture model essentially involved model selection. When the number of components was small, the model had fewer parameters. At this time, the model was simple and tended to underfit. When the number of components was large, the model had more parameters. At this time, the model was too complicated for datasets and tended to overfit. Model selection is beyond the scope of the paper and is not discussed here. We aimed to fully analyze how it affected the reconstruction results in our proposed method in this experiment.

In general, VAE could obtain good latent variable features and reconstruct its input well. We first used a lower latent variable dimension to show the improvement in reconstruction results with different numbers of components in our method. We set the latent variable dimension D equal to 2.

Figure 5(a) shows that some samples of the MNIST test data set we selected after training were used as reconstruction input. Figure 5(b) shows the reconstruction results using VAE. Figure 5(c)-(f) show the GMMpVAE for different number of components, where M represents the number of components in the Gaussian mixture model. It could be clearly seen that our reconstruction results were significantly better than VAE and M=1, even when M=2 or M=3 in the MNIST dataset.

However, since the low-dimensional latent variables were not enough to encode complex data sets like Omniglot, the encoder did not learn useful features, and the reconstruction results became abnormal as shown in Figure 6.

Then, we set the dimension D of the latent variable equal to 40. Figures 7, 8 and 9 show the reconstruction results when D=40 on the MNIST, Omnilot, and Frey Faces datasets, respectively.

In fact, the choice of M depended on the particular data set. On the one hand, as the number of components M increased, we can obtain better generative capabilities. At this time, the reconstruction effect was improving, and many details became visible even on the complex data set of Omniglot, but the larger M would increase the parameters of the entire network sharply.

E. LOG-LIKELIHOOD HISTOGRAM OF TEST SET

To understand the impact of our method on the model's generative ability more intuitively, we plotted the histogram of the LL of the test set. We selected 5000 samples from the test set of the MNIST and Omniglot datasets to plot the LL histogram.

As shown in Figure 10, we gave LL histograms of the test sets of the MNIST and Omniglot data sets, respectively. The results of each row represented VAE and our proposed method with the different number of components M. From Figure 10 we can see that the distribution of LL values was heavy-tailed or/and bimodal. One possible explanation for such features in the histogram could be that many samples were relatively simple to represent, while some were difficult (heavy-tailed). Comparing our proposed method with VAE, we can find that GMMpVAE not only had a better average effect but also generated more samples with lower LL values and fewer samples with higher LL values. At the same time, as the number of components in the Gaussian mixture model increased, the LL values of the test set became increasingly smaller.

VI. CONCLUSION

In the paper, we proposed a new VAE with optimizing Gaussian mixture model priors. We introduced the latent variable prior of the Gaussian mixture model and utilized its

aggregated posterior form to construct the KL distance of the prior and posterior of the mixture Gaussian model. Under the new framework, it was necessary to redefine the variational lower bound. We utilized the greedy algorithm to obtain an approximate solution of the KL distance, so as to obtain the approximate optimal ELBO solution. Our proposed method obtains state-of-the-art results on the MNIST, Omniglot, and Frey Faces datasets, compared to the generative models based on VAEs. At the same time, our detailed analysis shows that our proposed method can learn a better model.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arxiv preprint arXiv: 1312.6114, 2013.
[2] W. Wang, D. Yang, F. Chen, *et al.*, "Clustering With Orthogonal AutoEncoder," IEEE Access, vol. 7, pp. 62421-62432, 2019.
[3] K. Gregor, I. Danihelka, A. Graves, *et al.*, "Draw: A recurrent neural network for image generation" arXiv preprint arXiv:1502.04623, 2015.
[4] K. Gregor, I. Danihelka, A. Mnih, *et al.*, "Deep autoregressive networks," *International Conference on Machine Learning*, 2014, pp. 1242-1250.
[5] J. M. Tomczak, M. Welling, "VAE with a VampPrior," arXiv preprint arXiv:1705.07120, 2017.
[6] X. Chen, D. P. Kingma, T. Salimans, *et al.*, "Variational lossy autoencoder," arXiv preprint arXiv:1611.02731, 2016.
[7] M. Bauer, A. Mnih, "Resampled priors for variational autoencoders," arXiv preprint arXiv:1810.11428, 2018.
[8] J. Chou, "Generated Loss and Augmented Training of MNIST VAE," arXiv preprint arXiv:1904.10937, 2019.
[9] S. R. Bowman, L. Vilnis, O. Vinyals, *et al.*, "Generating sentences from a continuous space," arXiv preprint arXiv:1511.06349, 2015.
[10] S. Lin, S. Roberts, N. Trigoni, *et al.*, "Balancing Reconstruction Quality and Regularisation in ELBO for VAEs," arXiv preprint arXiv:1909.03765, 2019.
[11] J. Chou, G. Hathi, "Generated Loss, Augmented Training, and Multiscale VAE," arXiv preprint arXiv:1904.10446, 2019.
[12] B. Dai, D. Wipf, "Diagnosing and enhancing vae models," arXiv preprint arXiv:1903.05789, 2019.
[13] S. Zhao, J. Song, S. Ermon, "Infovae: Information maximizing variational autoencoders," arXiv preprint arXiv:1706.02262, 2017.
[14] D. J. Rezende, F. Viola, "Taming vaes," arXiv preprint arXiv:1810. 00597, 2018.
[15] D. J. Rezende, S. Mohamed, "Variational inference with normalizing flows," arXiv preprint arXiv:1505.05770, 2015.
[16] J. M. Tomczak, M. Welling, "Improving variational auto-encoders using householder flow," arXiv preprint arXiv:1611.09630, 2016.
[17] S. Ferdowsi, M. Diephuis, S. Rezaeifar, *et al.*, "$\rho $-VAE: Autoregressive parametrization of the VAE encoder," arXiv preprint arXiv:1909.06236, 2019.
[18] S. Cao, J. Li, K. P. Nelson, *et al.*, "Coupled VAE: Improved Accuracy and Robustness of a Variational Autoencoder," arXiv preprint arXiv:1906.00536, 2019.
[19] G. Safont, A. Salazar, L. Vergara, E. Gomez, V. Villanueva. "Probabilistic distance for mixtures of independent component analyzers," IEEE Transactions on Neural Networks and Learning Systems, vol. 29 no. 4, pp. 1161-1173, 2018.
[20] I. Tolstikhin, O. Bousquet, S. Gelly, *et al.*, "Wasserstein auto-encoders," arXiv preprint arXiv:1711.01558, 2017.
[21] M. Kim, Y. Wang, P. Sahu, *et al.*, "Relevance Factor VAE: Learning and Identifying Disentangled Factors," arXiv preprint arXiv:1902. 01568, 2019.
[22] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound," *Workshop in Adv. in Approximate Bayesian Inference (NIPS)*, vol. 1, 2016.
[23] M. Rosca, B. Lakshminarayanan, S. Mohamed, "Distribution matching in variational inference," arXiv preprint arXiv:1802.06847, 2018.
[24] D. J. Rezende, S. Mohamed, D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," arXiv preprint arXiv:1401.4082, 2014.
[25] A. Makhzani, J. Shlens, N. Jaitly, *et al.*, "Adversarial autoencoders," arXiv preprint arXiv:1511.05644, 2015.
[26] P. Goyal, Z. Hu, X, Liang, *et al.*, "Nonparametric variational auto-encoders for hierarchical representation learning," in *IEEE Int. Conf. on Computer Vision*, 2017, pp. 5094-5102.
[27] Y. Burda, R. Grosse, R. Salakhutdinov, "Importance weighted autoencoders," arXiv preprint arXiv:1509.00519, 2015.
[28] G. I. Liu, Y. Liu, M. Z. Guo, *et al.*, "Variational inference with Gaussian mixture model and householder flow," Neural Networks, vol. 109, pp. 43-55, 2019.
[29] J. R. Hershey, P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *32nd IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Honolulu, HI, Apr. 15-20, 2007, pp. 317-320.
[30] Y. Singer, M. K. Warmuth, "Batch and on-line parameter estimation of Gaussian mixtures based on the joint entropy," in *Proc. Adv. Neural Inf. Process. Syst*., 1999, pp. 578-584.
[31] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[32] Y. LeCun, L. Bottou, Y. Bengio, *et al.*, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1997.
[33] R. Salakhutdinov, I. Murray, "On the quantitative analysis of deep belief networks," *Proc. of the 25th Int. Conf. on Machine learning*, 2008, pp. 872-879.
[34] R. Salakhutdinov, J. B. Tenenbaum, A. Torralba, "Learning with hierarchical-deep models," *IEEE trans. on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1958-1971, 2012.
[35] Y. N. Dauphin, A. Fan, M. Auli, *et al.*, "Language modeling with gated convolutional networks," *Proc. of the 34th Int. Conf. on Machine Learning-Volume 70*, 2017, pp. 933-941.
[36] A. Oord, N. Kalchbrenner, K. Kavukcuoglu, "Pixel recurrent neural networks," arXiv preprint arXiv:1601.06759, 2016.
[37] X. Glorot, Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proc. of the 13th Int. conf. on artificial intelligence and statistics*, 2010, pp. 249-256.
[38] L. Dinh, D. Krueger, Y. Bengio, "Nice: Non-linear independent components estimation," arXiv preprint arXiv:1410.8516, 2014.
[39] T. Salimans, D. P. Kingma, M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," *Int. Conf. on Machine Learning*, 2015, pp. 1218-1226.
[40] D. Tran, R. Ranganath, D. M. Blei, "The variational Gaussian process," arXiv preprint arXiv:1511.06499, 2015.
[41] D. P. Kingma, T. Salimans, R. Jozefowicz, *et al.*, "Improved variational inference with inverse autoregressive flow," in *30th Conf. on Neural Inf. Process. Syst. (NIPS)*, Barcelona, SPAIN, 2016, pp. 4743-4751.
[42] J. T. Rolfe, "Discrete variational autoencoders," arXiv preprint arXiv:1609.02200, 2016.
[43] G. E. Hinton, S. Osindero, Y. W. The, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527-1554, 2006.
[44] L. Maaløe, C. K. Sønderby, S. K. Sønderby, *et al.*, "Auxiliary deep generative models," arXiv preprint arXiv:1602.05473, 2016.
[45] C. K. Sønderby, T. Raiko, L. Maaløe, *et al.*, "Ladder variational autoencoders," in *Proc. Adv. in neural inf. Process. Syst*., 2016, pp. 3738-3746.
[46] M. Bauer, A. Mnih, "Resampled priors for variational autoencoders," arXiv preprint arXiv:1810.11428, 2018.
[47] Y. Burda, R. Grosse, R. Salakhutdinov, "Accurate and conservative estimates of MRF log-likelihood using reverse annealing," *Artificial Intelligence and Statistics*, 2015, pp. 102-110.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2977671, IEEE Access

Chunsheng Guo is the associate Professor of Telecommunication Engineering, Hangzhou Dianzi University, China. He received the B.S. and M.S. and engaged in teaching and research at Anhui University of Technology in 1983 and 1997, respectively. He received a Ph.D. degree in 2002 in Communication and Information Systems from Nanjing University of Aeronautics and Astronautics, China. His research interests include video anomaly detection, pattern recognition, video moving object detection, and image analysis.

Jialuo Zhou studied for a master's degree in information and communication engineering in Hangzhou Dianzi University of Zhejiang, China in 2018. He is currently a member of the video analysis and processing team at Hangzhou Dianzi University. His research interests include pattern recognition and image analysis.

Huahua Chen is the associate Professor of Telecommunication Engineering, Hangzhou Dianzi University, China. He received the B.S., M.S. and Ph.D. in 1999, 2002 and 2005, respectively all in information and communication engineering from Zhejiang University, Hangzhou, China.

He joined the Hangzhou Dianzi University, in May 2005. His research interests include image processing, computer vision, video anomaly detection, and machine learning. He is the author of more than 30 research papers, and more than 10 inventions.

Na Ying is the associate Professor of Telecommunication Engineering, Hangzhou Dianzi University, China. She received her Ph.D. in 2006 in Communication Engineering from Jilin University, China. She has participated in about 10 scientific projects in the recent 5 years. She has co-authored more than 30 papers (collected by SCI / EI / ISTP) in international conference proceedings and journals, and was invited to give professional reports at the conference. She is Cochairman of the IEEE International Conference on Imaging Systems and Techniques.

Jianwu Zhang is the Professor of Telecommunication Engineering, Hangzhou Dianzi University, China. He received the B.S. and M.S. in 1983, 1988, respectively all in Wireless Communication Engineering from Nanjing Institute of Communication Engineering. He received a Ph.D. degree in 1999 in Information and Communication System from Zhejiang University, China. His research interests include Next Generation Mobile Telecommunication System, Network Optimization, Information security and AI application.

Di Zhou graduated from Southeast University in 1998 and received his MS degree from Southeast University in 2001. He is the president of Uniview Research Institute, engaged in the field of intelligent IoT for 19 years. He is the inventor of more than 400 authorized invention patents and 7 authorized US/EU invention patents, also the writer of two technology bestsellers The aria of a network: surveillance all-over-IP and Innovation 360: easy to explore creativity.