EXoN: EXplainable encoder Network

SeungHwan An*
Dept. of Statistics,
University of Seoul,
S. Korea

Jong-June Jeon
Dept. of Statistics,
University of Seoul,
S. Korea

Hosik Choi
Graduate School,
Dept. of Urban Big Data Convergence,
University of Seoul,
S. Korea

Abstract

We propose a new semi-supervised learning method of Variational AutoEncoder (VAE) which yields explainable latent space by EXplainable encoder Network (EXoN). The EXoN provides two useful tools for implementing VAE. First, we can freely assign a conceptual center of latent distribution for a specific label. We separate the latent space of VAE with multi-modal property of the Gaussian mixture distribution according to labels of observations. Next, we can easily investigate the latent subspace by a simple statistics, known as F-statistics, obtained from the EXoN. We found that both negative cross-entropy and Kullback-Leibler divergence play a crucial role in constructing explainable latent space and the variability of the generated samples from our proposed model depends on a specific subspace, called 'activated latent subspace'. With MNIST and CIFAR-10 dataset, we show that the EXoN can produce explainable latent space which effectively represents labels and characteristics of the images.

1 Introduction

In Variational AutoEncoder (VAE) [9], there are two main tasks of estimating a good generative model for high dimensional data, 1) constructing well-representing latent space for observations; and 2) recovering an original observation without loss of information. Since parameterized distributions are used for modeling probabilistic structure of observations and latent variables, the neural networks in VAE are given by a map from domain of observations to parameter space of latent variables and vice versa. The former neural network is referred to as an encoder and the latter one as a decoder.

VAE is fitted by the Variational Bayesian method [7] that maximizes Evidence of Lower Bound (ELBO). Since efficient sampling of latent variables is necessary for simulating the ELBO, standard Gaussian prior distribution is widely used as the latent distribution. However, the Gaussian assumption often causes failure to approximate a complex pattern of observations through the latent variables. [3, 22, 27, 2, 4] decompose the latent space by a mixture distribution. [24] proposes a VAE model in which the latent variables follow a mixture distribution with a hierarchical structure. The use of mixture distributions as latent structure improves the explainability of VAE. For example, the mixture latent structure of VAE can be more easily explained by a semi-supervised method [10, 15, 27].

However, the existing VAE models with the mixture latent distribution still have practical limitations. Before fitting the VAE, it is unknown which mixture component will be corresponding to a specific label of observations. In addition, it is difficult to impose any structural restriction on latent space, such as the proximity of latent features between some labeled data. As a result, we cannot design our latent space in which a specific region represents a feature of a specific labeled data of our interest with existing VAE models. This limitation of VAE leads to the following natural question "Is it possible to construct a feature space where two different observations are effectively interpolated?"

*Official github repository: `https://github.com/an-seunghwan/EXoN`

Preprint. Under review.

On the basis of our question, we propose a new semi-supervised VAE model. We employ the mixture Gaussian as latent distribution and focus on constructing an explainable encoder of the VAE. In our model, each component of the mixture distribution corresponds to a specific label. The neighborhood of each location parameter of the mixture distribution can be regarded as a set of latent points generating observations with a specific label. Thus, we can generate observations on a line segment interpolating two points in the latent space. In practical view, the main advantage of our model is that its latent space can be pre-designed by the encoder.

In addition, we can explain the latent space of our VAE model by a post-ad-hoc method: mapping an observation to the latent variable and reconstructing the observation by a sample from the neighborhood of the mapped image. Since each observation has its own representation on the latent space by the proposed encoder [28], we can easily compare latent features by investigating a set of observations of interest. By this post-ad-hoc method, we discover that only a part of latent space determines characteristics of generated samples. That is, the encoder of our model selectively activates latent subspace and the generated samples from our model substantially depends on the values of the subspace. Thus, we call the encoder of our proposed VAE model EXoN (EXplainable encoder Network) by borrowing the term in the field of gene biology. Figure 1 shows a series of images obtained by interpolating two points on the latent subspace which are detected by EXoN.

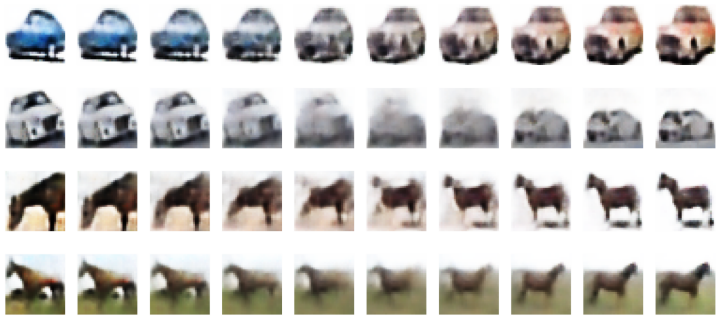

Figure 1: Examples of interpolated image in CIFAR-10 dataset.

This paper is organized as follows. Section 2 briefly introduces Variational AutoEncoder, and Section 3 proposes our VAE model with a mixture prior including its derivation. Section 4 shows the results of numerical simulations. Concluding remarks follow in Section 5.

2 Preliminary

2.1 Variational AutoEncoder for Unsupervised Learning

Let $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m$ and $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^d$ with $d < m$ be the observed data and the latent variable of VAE model, and denote the associated probability density functions (pdf) by $p(\mathbf{x})$ and $p(\mathbf{z})$. Let $p(\mathbf{x}|\mathbf{z})$ be the conditional pdf of $\mathbf{x}$ for a given $\mathbf{z}$, and assume that $\mathbf{x}|\mathbf{z} \sim N(D(\mathbf{z}), \beta \cdot \mathbf{I})$ where $D : \mathcal{Z} \mapsto \mathcal{X}$, $\mathbf{I}$ is the $m \times m$ identity matrix and $\beta > 0$. $D(\mathbf{z})$ is parameterized as a neural network model with parameter θ, $D(\mathbf{z}; \theta)$. The conditional pdf, $p(\mathbf{x}|\mathbf{z}; \theta, \beta)$, is referred as the decoder of VAE. We also use the term of decoder as the map $\mathbf{z} \mapsto (D(\mathbf{z}; \theta), \beta)$, because $p(\mathbf{x}|\mathbf{z}; \theta, \beta)$ is fully parameterized by $(D(\mathbf{z}; \theta), \beta)$.

The parameter of VAE (θ, β) is estimated by Variational method [9, 19] that maximizes ELBO (Evidence of Lower Bound). The ELBO of $\log p(\mathbf{x}; \theta, \beta)$ is obtained from the inequality

$$\log p(\mathbf{x}; \theta, \beta) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}; \theta, \beta)] - \mathcal{KL}\left(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right), \tag{1}$$

where $\mathcal{KL}(q|p)$ denotes Kullback-Leibler divergence from p to q. If the support of $p(\mathbf{z})$ is larger than that of $q(\mathbf{z}|\mathbf{x})$ for all $\mathbf{x}$, (1) is always valid for any $q(\mathbf{z}|\mathbf{x})$. [9, 19] introduce a neural network with parameter ϕ for modeling $q(\mathbf{z}|\mathbf{x}; \phi)$, and obtain the ELBO objective for finite samples $x_1, \cdots, x_n$:

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{z}|x_i;\phi)}[\log p(x_i|\mathbf{z}; \theta, \beta)] - \mathcal{KL}\left(q(\mathbf{z}|x_i; \phi) \| p(\mathbf{z})\right). \tag{2}$$

Here, $q(\mathbf{z}|\mathbf{x}, \phi)$ is referred to as the encoder of VAE or the posterior of a latent distribution. Multivariate Gaussian distributions are widely used as $q(\mathbf{z}|\mathbf{x}, \phi)$ in which the mean and covariance are modeled by a neural network. We denote the parameter of the neural network by ϕ, and thus the parameters of VAE consist of two terms, ϕ in the encoder and (θ, β) in the decoder.

In practice, (2) is approximated by

$$-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\beta}\Big(\frac{1}{2}\|x_i - D(z_i;\theta)\|^2 + \beta \cdot \mathcal{KL}(q(\mathbf{z}|x_i;\phi)\|p(\mathbf{z}))\Big) - \frac{m}{2}\log 2\pi\beta \tag{3}$$

where z_i is a sample from $q(\mathbf{z}|x_i;\phi)$ and $\|\cdot\|$ is l_2-norm, and VAE is fitted by maximizing (3) with respect to (θ, β, ϕ) [14]. The first term in (3) is related to the precision of generated samples. Since the KL-divergence is always non-negative, [5, 17, 13] explained β as a tuning parameter which controls the regularization of θ and ϕ.

2.2 Variational AutoEncoder for Semi-Supervised Learning

Let $\mathbf{y} \in \mathcal{Y} = \{1, \cdots, K\}$ be a discrete random variable and let the joint probability density of $(\mathbf{x}, \mathbf{y})$ be $p(\mathbf{x}, \mathbf{y}; \theta, \beta)$. We consider a semi-supervised learning problem where $\mathbf{y}$ denotes a label of $\mathbf{x}$. Suppose that we observe n_1 labeled and n_2 unlabeled random samples from $p(\mathbf{x}, \mathbf{y}; \theta, \beta)$ and $p(\mathbf{x}; \theta, \beta)$. Denote the sets of indices for labeled and unlabeled samples by I_l and I_u, and $n = n_1 + n_2$. Then, the MLE of (θ, β) maximizes

$$\frac{1}{n}\sum_{i\in I_l}\log p(x_i, y_i; \theta, \beta) + \frac{1}{n}\sum_{i\in I_u}\log p(x_i; \theta, \beta), \tag{4}$$

where (x_i, y_i) and x_i are random samples from $p(\mathbf{x}, \mathbf{y}; \theta)$ and $p(\mathbf{x}; \theta)$, respectively. The semi-supervised VAE model to estimate θ in (4) is firstly proposed by [10]. The ELBO of (4) is derived from the joint probability as the following inequality:

$$\log p(\mathbf{x}, \mathbf{y}; \theta) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y};\phi)}[\log p(\mathbf{x}|\mathbf{y}, \mathbf{z}; \theta) + \log p(\mathbf{y}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}, \mathbf{y}; \phi)]. \tag{5}$$

Here, it is assumed that $q(\mathbf{z}|\mathbf{x}, \mathbf{y}; \phi)$ is a multivariate normal distribution of which location and scale parameters are modeled by a neural network with parameter ϕ. Note that the location parameter of $q(\mathbf{z}|\mathbf{x}, \mathbf{y}; \phi)$ is a map from $\mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Z}$. For simplicity, denote the right term of (5) by $C(\mathbf{x}, \mathbf{y}; \theta, \phi)$.

By marginalization of (5), $\log p(\mathbf{x}; \theta) \geq \sum_{\mathbf{y}\in\mathcal{Y}} q(\mathbf{y}|\mathbf{x}; \phi)C(\mathbf{x}, \mathbf{y}; \theta, \phi) + \mathcal{H}(q(\mathbf{y}|\mathbf{x}; \phi))$, where $\mathcal{H}(q)$ is entropy of q, and thus the ELBO of (4) is given by

$$\frac{1}{n}\sum_{i\in I_l} C(x_i, y_i; \theta, \phi) + \frac{1}{n}\sum_{i\in I_u}\left(\sum_{\mathbf{y}\in\mathcal{Y}} q(\mathbf{y}|x_i; \phi)C(x_i, \mathbf{y}; \theta, \phi) + \mathcal{H}(q(\mathbf{y}|x_i; \phi))\right). \tag{6}$$

Since $C(x_i, y_i; \theta, \phi)$ is not of closed form, (6) is approximated by Monte-Carlo method with random samples z_i from $q(\mathbf{z}|x_i, y_i; \phi)$ for $i \in I_l$.

(6) is the sum of two lower bounds of $\log p(x_i, y_i; \theta)$ and $\log p(x_i; \theta)$, which are linked with $q(\mathbf{z}|\mathbf{x}, \mathbf{y}; \phi)$. The latent space can be explained by the visualization of a random sample z_i from $q(\mathbf{z}|x_i, y_i; \phi)$ which shows a pattern of the latent structure corresponding to each label. However, it is not guaranteed that the latent space would be explained according to each label or the characteristics of observations. Motivated with this concern, we propose the VAE model of which the latent structure is able to be pre-designed by a specific encoder. The encoder employs a mixture distribution of which a modal corresponds to each label.

3 Proposed Model

3.1 Model Assumptions

We propose a new VAE model with explainable latent space where we can freely assign a conceptual center of a latent distribution for a specific label. First, we set a latent distribution for each label. Let $\mathbf{z}|\mathbf{y} = k \sim N(\mu_k^0, diag\{(\sigma_k^0)^2\})$ and $p(\mathbf{y} = k) = w_k^0$ for $k \in \mathcal{Y}$ as a prior distribution.

The distribution of $\mathbf{z}$ is written by $p(\mathbf{z}) = \sum_{k=1}^{K} w_k^0 \cdot \mathcal{N}(\mathbf{z}|\mu_k^0, diag\{(\sigma_k^0)^2\})$. Here, $\mu_k^0 \in \mathbb{R}^d$ and $\sigma_k^0 \in \mathbb{R}_+^d$ are pre-determined parameters denoting the conceptual center and dispersion of the latent variable for each label, and we call the choice of μ_k^0 and σ_k^0 the pre-design of latent space for labeled data. [17] accounts for the use of mixture distribution as a customized design of the latent space. Since μ_k^0 and w_k^0 for all k are fixed, we omit the notation of the parameters.

We assume that the decoder $p(\mathbf{x}|\mathbf{y} = k, \mathbf{z})$ is a Gaussian distribution with mean $\mu_k(\mathbf{z})$ and common covariance $\beta \cdot \mathbf{I}$. Then, the conditional distribution of $\mathbf{x}$ for a given $\mathbf{z}$ is followed by $p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{y} = k|\mathbf{z}) \cdot \mathcal{N}(\mathbf{x}|\mu_k(\mathbf{z}), \beta \cdot \mathbf{I})$, which is a mixture Gaussian distribution with K components. However, if we let sufficiently separated μ_k^0s from each other and small σ_k^0s, the probability mass of $p(\mathbf{y}|\mathbf{z})$ will be concentrated on a single point and the generation process mainly depends on $\mathcal{N}(\mathbf{x}|\mu_k(\mathbf{z}), \beta \cdot \mathbf{I})$ for some k with the highest probability of $p(\mathbf{y} = k|\mathbf{z})$. For each k, we can consider the subset on latent space, $\{\mathbf{z} \in \mathcal{Z} : k = \arg\max_j p(\mathbf{y} = j|\mathbf{z})\}$, which can be regarded as the feature space describing the generation of $\mathbf{x}$ with the label k. By this pre-designed latent space, we can approximate $p(\mathbf{x}|\mathbf{z})$ as a unimodal Gaussian distribution $\mathcal{N}(D(\mathbf{z};\theta), \beta \cdot \mathbf{I})$ where $D(\mathbf{z};\theta)$ is a neural network from $\mathcal{Z}$ to $\mathcal{X}$. That is, the decoder of the proposed VAE is given by $p(\mathbf{x}|\mathbf{z};\theta,\beta) = \mathcal{N}(\mathbf{x}|D(\mathbf{z};\theta), \beta \cdot \mathbf{I})$.

Our model assumptions imply that the encoder $p(\mathbf{z}|\mathbf{x})$ is given by a mixture distribution, $p(\mathbf{z}|\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{y} = k|\mathbf{x}) p(\mathbf{z}|\mathbf{y} = k, \mathbf{x})$ where $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{y}, \mathbf{x})$ depend on (θ, β). However, since the closed form of $p(\mathbf{z}|\mathbf{y} = k, \mathbf{x})$ is unknown, $p(\mathbf{z}|\mathbf{x})$ is computationally prohibitive. Thus, we apply approximation method by introducing a classification model $w(\mathbf{y}|\mathbf{x};\eta)$ and proposal distributions $g(\mathbf{z}|\mathbf{y} = k, \mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi), diag\{(\sigma_k(\mathbf{x};\xi))^2\})$ for $p(\mathbf{y} = k|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{y} = k, \mathbf{x})$, respectively. The proposal distributions are multinomial and Gaussian distribution with a diagonal covariance whose parameters are modeled by neural networks. Let the parameters of the neural networks be η and ξ. Finally, posterior distribution is approximated by

$$q(\mathbf{z}|\mathbf{x};\eta,\xi) = \sum_{k=1}^{K} w(\mathbf{y} = k|\mathbf{x};\eta) g(\mathbf{z}|\mathbf{y} = k, \mathbf{x};\xi) \tag{7}$$

We can write the parameterized model of the encoder by a map from $\mathcal{X}$ to $\left((0,1) \times \mathbb{R}^d \times \mathbb{R}_+^d\right)^K$ which denotes K cartesian product space corresponding to $w(\mathbf{y} = k|\mathbf{x};\eta)$, $\mu_k(\mathbf{x};\xi)$ and $\sigma_k(\mathbf{x};\xi)$ for $k = 1, \cdots, K$.

3.2 EXoN: Semi-Supervised VAE

Our proposed VAE model is derived from the joint probability density function $p(\mathbf{x}, \mathbf{y})$. We decompose $p(\mathbf{x}, \mathbf{y})$ into $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$, and apply the derivation of the ELBO only to $\log p(\mathbf{x})$ as (1). In addition, we approximate $p(\mathbf{y}|\mathbf{x})$ by $w(\mathbf{y}|\mathbf{x};\eta)$, the proposal classification model. Let $p(\mathbf{x}, \mathbf{y};\theta,\beta)$ be parameterized joint distribution of $(\mathbf{x}, \mathbf{y})$, then our approximated ELBO is given by

$$\begin{aligned}
& \log p(\mathbf{x}, \mathbf{y};\theta,\beta) \\
= \; & \log p(\mathbf{x};\theta,\beta) + \log p(\mathbf{y}|\mathbf{x};\theta,\beta) \\
\geq \; & \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\eta,\xi)}[\log p(\mathbf{x}|\mathbf{z};\theta,\beta)] - \mathcal{KL}\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right) + \log p(\mathbf{y}|\mathbf{x};\theta,\beta) \\
\simeq \; & \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\eta,\xi)}[\log p(\mathbf{x}|\mathbf{z};\theta,\beta)] - \mathcal{KL}\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right)}_{\text{(ii)}} + \underbrace{\log w(\mathbf{y}|\mathbf{x};\eta)}_{\text{(i)}}.
\end{aligned} \tag{8}$$

The first term (i) in (8) is the typical ELBO used in the unsupervised VAE learning, and the remaining term (ii) is the negative cross-entropy. Note that (i) and (ii) are coupled with the shared parameter η. The negative cross-entropy plays roles with a classifier for $\mathbf{y}$ as well as a posterior probability of a latent variable to the pre-designed mixture component. Therefore, $w(\mathbf{y}|\mathbf{x};\eta)$ with lower classification error can separate the latent space more clearly by $q(\mathbf{z}|\mathbf{x};\eta,\xi)$, which will be shown in Section 4.1.

$\mathcal{KL}\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right)$ of (8) is not written by a closed form, so we use an upper bound of the KL-divergence [22, 4], $\mathcal{KL}^u\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right)$ (see the Appendix A.1). It follows that

$$\log p(\mathbf{x}, \mathbf{y};\theta,\beta) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x};\eta,\xi)}[\log p(\mathbf{x}|\mathbf{z};\theta,\beta)] - \mathcal{KL}^u\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right) + \log w(\mathbf{y}|\mathbf{x};\eta), \tag{9}$$

whenever $p(\mathbf{y}|\mathbf{x};\theta,\beta) = w(\mathbf{y}|\mathbf{x};\eta)$. From (9), the ideal objective function of VAE is straightforwardly defined as

$$\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[\log w(\mathbf{y}|\mathbf{x};\eta)] + \mathbb{E}_{p(\mathbf{x})}\Big[\mathbb{E}_{q(\mathbf{z}|\mathbf{x};\eta,\xi)}[\log p(\mathbf{x}|\mathbf{z};\theta,\beta)] - \mathcal{KL}^u\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right)\Big]. \tag{10}$$

Thus, an objective function under finite samples is naturally given by

$$\mathcal{L}(\theta, \beta, \xi, \eta) + \frac{1}{|I_l|} \sum_{i \in I_l} \log w(y_i|x_i; \eta), \tag{11}$$

where $\mathcal{L}(\theta, \beta, \xi, \eta) = \frac{1}{|I_l \cup I_u|} \sum_{i \in I_l \cup I_u} \log p(x_i|z_i; \theta, \beta) - \mathcal{KL}^u \left(q(\mathbf{z}|x_i; \eta, \xi) \| p(\mathbf{z})\right)$, z_i is a random sample from $q(\mathbf{z}|x_i; \eta, \xi)$ and $|A|$ is the cardinality of a set A.

We consider a regularization of η in the encoder and β in the decoder. By adding $\lambda_1 \cdot \frac{1}{|I_l|} \sum_{i \in I_l} \log w(y_i|x_i; \eta)$ for $\lambda_1 \geq 0$ to (11), we forces $\mathcal{L}(\theta, \beta, \xi, \eta)$ to be loosely maximized under finite samples. Since $\log w(y|\mathbf{x}; \eta)$ is always negative, it plays a role as a penalty function in the maximization problem. Also, we consider a regularization method for β by introducing the penalty function $-\lambda_2/\beta$ for $\lambda_2 \geq 0$ which prevents β from going to zero. Thus, the proposed VAE model is defined by maximizing

$$\mathcal{L}_{\lambda_1, \lambda_2}(\theta, \beta, \xi, \eta) = \mathcal{L}(\theta, \beta, \xi, \eta) + (\lambda_1 + 1) \cdot \frac{1}{|I_l|} \sum_{i \in I_l} \log w(y_i|x_i; \eta) - \frac{\lambda_2 m}{2\beta}. \tag{12}$$

If λ_1 is set to be large, the latent space tends to be well separated according to labels for a fixed λ_2. This separation is affected by the KL-divergence term as well as the negative cross-entropy, because the shared model parameter η in the two terms is learned toward reducing the KL-divergence $\mathcal{KL}^u \left(q(\mathbf{z}|\mathbf{x}; \eta, \xi) \| p(\mathbf{z})\right)$. [10] first employed the cross-entropy as a penalty function of the VAE and [15, 26, 20, 25] use the same penalty function in the subsequent papers. [13] referred to the necessity of the negative cross-entropy restricting lower bound of the objective of VAE to obtain disentangled latent representation. In our semi-supervised VAE model, the penalty function is applied with a similar idea, but the derivation of the regularized objective function stems from (10) unlike the existing studies.

When λ_2 is large, the generated samples tend to be overlapped in a sense that variations of the samples are reduces. When the other parameters except for β are fixed, the optimal solution of β is given by

$$\hat{\beta} = \frac{1}{|I_l \cup I_u|} \sum_{i \in I_l \cup I_u} \frac{1}{m} \|x_i - D(z_i; \theta)\|^2 + \lambda_2,$$

z_i is a random sample from $q(\mathbf{z}|x_i; \eta, \xi)$. The solution is the variance parameter in the decoder, which denotes the variation of a generated samples. If $\hat{\beta}$ is large, the possibility of multiple observations overlapping increases. In this case, each point in the latent space loses its own representation for a specific observation.

Furthermore, a large λ_2 indirectly increases the weight of the KL-divergence term in our objective function. Denote the random vector associated with the distribution of the kth component of the mixture distribution $p(\mathbf{z})$ by $\mathbf{z}^k = (\mathbf{z}_1^k, \cdots, \mathbf{z}_d^k)^\top \sim N(\mu_k^0, diag\{(\sigma_k^0)^2\})$. When (12) is dominated by $\mathcal{KL}^u \left(q(\mathbf{z}|\mathbf{x}; \eta, \xi) \| p(\mathbf{z})\right)$ and the classification error of $w(\mathbf{y}|\mathbf{x}; \eta)$ is almost zero, $\log \mathbb{E}_{\mathbf{x}|\mathbf{y}=k} \left[\frac{\text{Var}(\mathbf{z}_j^k)}{\text{Var}(\mathbf{z}_j^k|\mathbf{x}; \xi)} \right] \simeq 0$ for all j and k, where $p(\mathbf{x}|\mathbf{y} = k; \eta) = w(\mathbf{y} = k|\mathbf{x}; \eta) p(\mathbf{x}) / p(\mathbf{y} = k)$.

Meanwhile, the relaxation of the KL-divergence increases $\text{Var}_{\mathbf{x}|\mathbf{y}=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x}; \xi)]$ for some j and k. In our numerical study, we found that the subspace

$$\{j \in \{1, \cdots, d\} : \frac{\text{Var}_{\mathbf{x}|\mathbf{y}=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x}; \xi)]}{\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x}; \xi)]} > \delta\}$$

for some positive δ represents informative characteristics of generated observations [18]. Thus, we call the encoder of the proposed VAE the EXplainable encoder Network (EXoN) and call this latent subspace 'activated latent subspace'. Details are discussed in Section 4.2. Following theorem shows that this relaxation of the KL-divergence produces the activated latent subspace.

Theorem 1. *Let $p(\mathbf{z})$ the mixture prior distribution defined in Section 3.1 and let $q(\mathbf{z}|\mathbf{x}; \eta, \xi)$ be* (7)*. Then,*

$$\mathbb{E}_{\mathbf{x}} \mathcal{KL}^u \left(q(\mathbf{z}|\mathbf{x}; \eta, \xi) \| p(\mathbf{z})\right) \leq \mathbb{E}_{\mathbf{x}} \mathcal{KL}(w(\mathbf{y}|\mathbf{x}; \eta) \| p(\mathbf{y})) + \sum_{k=1}^{K} \sum_{j=1}^{d} \frac{w_k^0}{2} \log \mathbb{E}_{\mathbf{x}|\mathbf{y}=k} \left[\frac{Var(\mathbf{z}_j^k)}{Var(\mathbf{z}_j^k|\mathbf{x}; \xi)} \right].$$

4 Experiments

We ran all experiments using Geforce GTX 2080 Ti GPU and 16GB RAM and our experimental codes are all available with tensorflow [1].

4.1 MNIST dataset

We use the MNIST dataset [12] to fit our VAE model and all values of the images are scaled to the range of $(0, 1)$. We consider the 2-dimensional latent space and let the decoder be two-layered neural network with 200 and 400 hidden nodes where the activation function in the neural network is the leaky ReLU with the slope parameter 0.01. The encoder of our VAE also comprises two layer neural network with 200, 400 hidden nodes and describes the parameters 10-component Gaussian mixture distribution with mixing probability, mean vector, diagonal covariance elements. Thus, the encoder is a map from $\mathcal{X}$ to $\left((0,1) \times \mathbb{R}^2 \times \mathbb{R}_+^2\right)^{10}$.

RMSprop [21] with a learning rate of 0.001, and Gumbel-Softmax approximation method [6, 16] are applied to fitting the model. The temperature hyperparameter of Gumbel-Softmax method is fixed to 1 during the training iterations. We randomly divide the MNIST train dataset into unlabeled with 55000 observations and labeled with 5000 observations, and ensure that labeled dataset has the same number of observations belonging to each class. We set the mini-batch size to be 2000 for unlabeled and 179 for labeled dataset, and iterated 200 epochs.

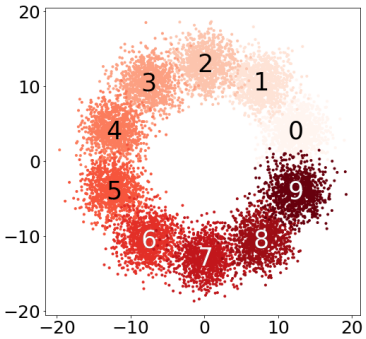

Figure 2: Scatter plot of sampled latent variables from a 10-component mixture Gaussian prior distribution; labels of each cluster are annotated and distinguished by color.

Figure 2 illustrates the random samples from the pre-designed prior distribution. The labels corresponding to the components are assigned counterclockwise from 3 0'clock. The prior's parameters are $\mu_k^0 = r \cdot (\cos(\frac{\pi}{10}k), \sin(\frac{\pi}{10}k))^\top$ for $k = 1, \cdots, 10$, where $r = 4/\sin(\frac{\pi}{10})$ and $\sigma_k^0 = (4, 4)^\top$.

To evaluate the fitted VAE model, we use three types of measurements, the average negative single-scale structural similarity (negative SSIM) [23, 27], the classification error and the KL-divergence. We compute the negative SSIM to measure diversity of generated images by our model and it is measured on the generated samples produced by sampled latent variables. And the classification error and the KL-divergence is measured using test dataset (see the Appendix A.3.1 for details).

We investigate the fitted model performance by selecting the tuning parameters in (12). The measurements are evaluated according to λ_1 and λ_2, and the heat map results are displayed in the Appendix A.3.1. We chose our VAE model's final tuning parameter pair (λ_1, λ_2) as (100, 1.1) such that the fitted model shows a large negative SSIM (high diversity), a low classification error and a low KL-divergence value. Figure 3 illustrates fitted results from the model with chosen tuning parameters.

The first panel of Figure 3 visualizes that the fitted model's posterior distribution is regularized to the pre-designed structure of the prior (see Figure 2). The second panel displays how observations are generated from the latent space and matched to each latent point. Generated images in the center of the second panel do not indicate any label because there is no cluster in the center of the pre-designed latent space. Specifically, images simulated from the area between mixture components are in the form of mixed labels corresponding to each cluster. This result implies that our model can give manual interpolation results in accord with the design of latent space. The third panel shows how

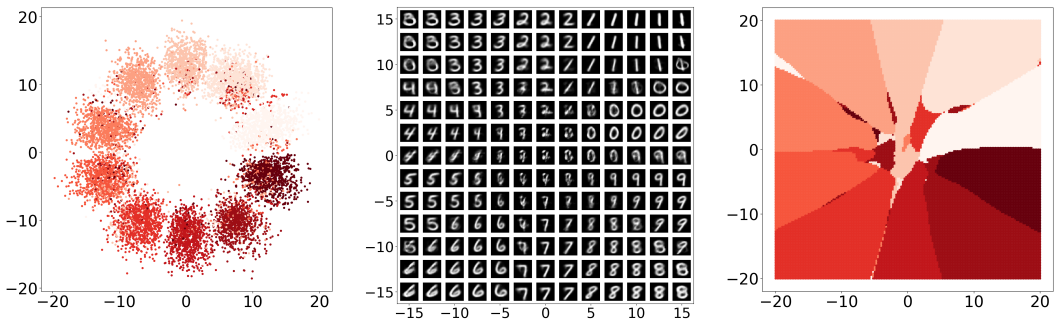

Figure 3: From left to right, 1) scatter plot of $z_i \sim q(\mathbf{z}|x_i)$ where x_i is an observation in the test dataset; 2) generated images from grid points on the latent space, and 3) labels of maximum conditional probabilities $w(y|\mathbf{x};\eta)$ given reconstructed images which are produced by grid points; labels are distinguished by color and all results are produced from the fitted VAE model with tuning parameter pair (100, 1.1).

labels of observations which are generated from more denser grid points are distributed on the latent space. It shows that the latent space is continuously separated by labels, even outside the posterior distribution.

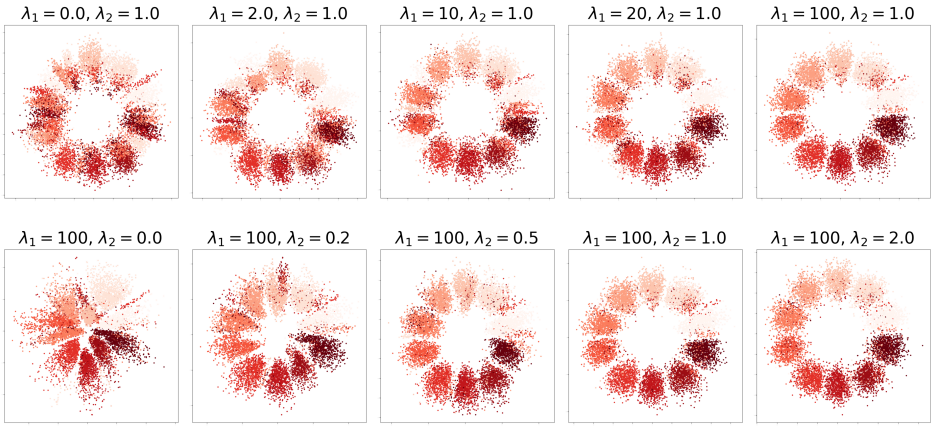

Figure 4: Scatter plot of latent variables sampled from the fitted posterior distribution, given test dataset with various tuning parameter pairs.

[14] shows that careful selection of the observation variance β is necessary to construct the decoder of VAE. Motivated by this paper, we investigate the role of negative cross-entropy and the regularization of $1/\beta$ in our VAE model. Each effect of two regularizers in (12) is displayed in Figure 4. Because λ_1 penalizes negative cross-entropy, larger λ_1 brings more exact accordance between an assigned cluster of latent variable and given true label so that mixture components of the posterior are separated as those of the prior. And larger λ_2 penalizes shrinkage of β, larger λ_2 indirectly makes the KL-divergence term dominant in (12) and posterior distribution structure become similar to that of the prior.

4.2 CIFAR-10 dataset

We also applied our proposed VAE model to the CIFAR-10 dataset [11] and all values of the images are scaled to the range of $(0, 1)$. 300-dimensional latent space is used and network architectures of the VAE model are shown in the Appendix A.4.3. The pre-designed prior mean vectors are divided into 100-dimensional label-relevant and 200-dimensional label-irrelevant parts [27]. 10 label-relevant mean vector parts lie on a line segment from $6 \cdot \mathbf{1}^{100}$ to $-6 \cdot \mathbf{1}^{100}$ and label-irrelevant mean vector

parts are set to be all zero. The diagonal elements of the prior covariance matrix are divided as same with the mean vectors, 0.1 for label-relevant and 1.0 for label-irrelevant dimensions.

Adam [8] with a learning rate of 0.005 is applied to fitting the model. We chose λ_1 to be 100, and the temperature parameter of Gumbel-Softmax method is fixed 1 during training iterations. We randomly divide the train dataset into 40000 unlabeled and 10000 labeled observations, and ensure that labeled dataset has the same number of observations belonging to each class. We set the mini-batch size to be 128 and iterated 200 epochs. Since the CIFAR-10 dataset is more complicated to classify than the MNIST dataset and 300-dimensional large latent variable size, it is hard to fit our posterior structure to the pre-designed latent space. So, we pre-trained parameter η with the CIFAR-10 labeled dataset. In classifier pre-training, Adam with a learning rate 0.002 is used. And we set the mini-batch size to be 128 and iterated 15 epochs.

In here, we modified our covariance matrix assumption of the decoder. Instead of a diagonal covariance matrix with equal β elements, we modeled diagonal elements of covariance matrix as a function of mean parameter; $diag\{D(\mathbf{z};\theta)\cdot(\mathbf{1}-D(\mathbf{z};\theta))\}$ where $\mathbf{1}\in\mathbb{R}^m$. However, $D(\mathbf{z};\theta)$ is trained to be close to $\mathbf{x}$ and for simplicity of modeling, we used the weighted Gaussian model (it is also same with second order taylor expansion of binary cross-entropy):

$$\log p(\mathbf{x}|\mathbf{z};\theta,\beta) = -\frac{1}{2}\sum_{j=1}^{m}\frac{1}{\mathbf{x}_j(1-\mathbf{x}_j)+\epsilon}(\mathbf{x}_j - D(\mathbf{z};\theta)_j)^2 - \frac{1}{2}\sum_{j=1}^{m}\log(2\pi(\mathbf{x}_j(1-\mathbf{x}_j)+\epsilon)) \quad (13)$$

where ϵ prevents denominator of weight to become zero and controls maximum value, and second constant term is not included in training. As the activated latent subspace depends on the KL-divergence contained in ELBO, we controlled the diversity of the feature characteristics that the latent space represents via ϵ. Because ϵ plays the similar role of tuning parameter λ_2, so it indirectly controls how much the KL-divergence is minimized in our VAE model [18]. This is shown in following Figures 6 and 7. Of course, the observation variance parameter β is not used in this case, so λ_2 is zero. We expect that the modified covariance matrix would help our decoder capturing more complicated informations of observations [14].

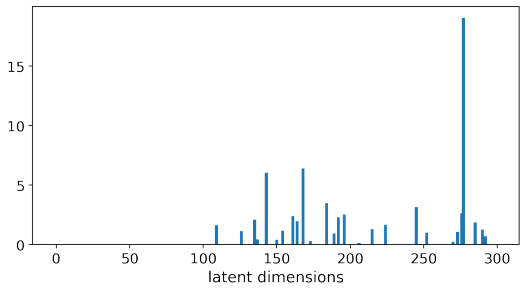

Figure 5: Plot of the activated latent subspace measurement of automobile class.

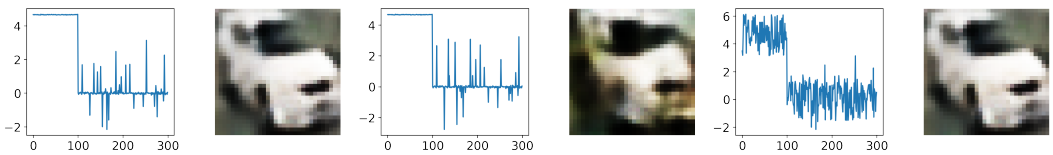

Figure 6: Visualization of pairs which are a latent variable and its reconstructed image; from left to right, 1) the posterior mean parameter $\mu_k(x;\xi)$, 2) uniform noises are added on the activated latent subspace, and 3) uniform noises are added except for the activated latent subspace; noises are sampled from $U(-1.5, 1.5)$, x is an arbitrary sample and k is automobile class.

As we mentioned at Introduction and Section 3.2, we investigated the activated latent subspace. Both Figure 5 and Figure 6 are obtained from the fitted model where ϵ is 0.02. From Figure 5, we observed 27 activated dimensions with threshold 0.1. Figure 6 shows that only the activated latent subspace affects features of generated samples. Therefore, the activated latent subspace should be considered carefully to explain the latent space and interpolate latent variables.

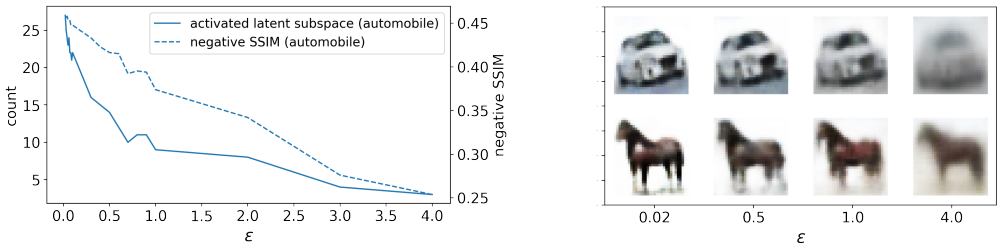

Figure 7: From left to right, 1) plot of the number of activated latent subspace and negative SSIM value of reconstructed images produced by $D(z_i;\theta)$ where $z_i = \mu_k(x_i;\xi)$ where x_i is an observation in the training dataset of automobile class k, and 2) reconstructed images produced by $D(z;\theta)$ where $z = \mu_k(x;\xi)$, for an arbitrary sample x belonging automobile or horse class, with respect to various ϵs.

Left panel of Figure 7 indicates that ϵ controls the activated latent subspace, and the diversity of observation's characteristics increases as ϵ increases. And this result is also shown in right panel qualitatively. Lastly, Figure 1 in Introduction is given by $(t, D(z(t);\theta)), 0 \leq t \leq 1$ where $z(t) = t \cdot \mu_k(x;\xi) + (1-t) \cdot \mu_k(x';\xi)$ and arbitrary different samples x, x' which are belong to same class k.

5 Conclusion

This paper proposed a new semi-supervised learning method with Variational AutoEncoder when using a mixture Gaussian distribution as a prior distribution structure. We derived the objective ELBO by decomposing the joint distribution of data and label into marginal data and conditional distribution of label. In this way, our proposed method relaxed computational issues and derived a negative cross-entropy regularizer, naturally which is required for efficient model training. Additionally, we introduced the penalty function for the purpose of preventing the shrinkage issue of observation model noise.

Through numerical experiments, the proposed method reveals the effects of these regularizers. First, the negative cross-entropy regularizer controls the correspondence between an assigned cluster and a true given label in the latent variable sampling process from the posterior. So that mixture components of the posterior are decomposed sufficiently. Next, penalty function of the observation noise regularizes the posterior distribution structure as the pre-designed latent space. A practical advantage of our method is that the manually interpolated result corresponds to the user's specific purpose and can be obtained by configuring the prior distribution structure in advance. So, the proposed method will be useful when a particular interpolation pattern is needed or a model needs to be trained with only a few labeled data.

In addition, we introduced new concepts on KL-divergence minimizing and its relationship with latent space interpretation using activated dimension. By real-data experiment, we revealed that data information is concentrated in a very small region of latent space and only activated dimensions are responsible for reconstructing given datapoint. We hope that our approach could help understanding how the fitted latent space is constructed.

The proposed VAE provides a natural explanation for using a cross-entropy penalty, unlike the conventional VAE [10]. We decompose the joint probability model into two terms, the marginal probability of predictor and the conditional probability of label for predictor, and apply derivation of ELBO of the VAE only to the marginal probability model. Then, the cross-entropy is naturally derived from the conditional probability. It plays a role in restricting the feature space from being well separated according to the prior with a pre-designed mixture distribution. The marginal and conditional probability model shares the feature space of the VAE. Also, it is unnecessary to use the joint probability for obtaining the ELBO, and thus computation of the fitting VAE model is more efficient than the conventional VAE.

Future works: infinite class, continuous mixture distribution distribution, efficient computation

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Yasemin Bozkurt Varolgunes, Tristan Bereau, and Joseph F Rudzinski. Interpretable embeddings from molecular simulations using gaussian mixture variational autoencoders. *arXiv e-prints*, pages arXiv–1912, 2019.

[3] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[4] Chunsheng Guo, Jialuo Zhou, Huahua Chen, Na Ying, Jianwu Zhang, and Di Zhou. Variational autoencoder with optimizing gaussian mixture model priors. *IEEE Access*, 8:43992–44005, 2020.

[5] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[6] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[7] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[12] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[13] Yang Li, Q. Pan, Suhang Wang, Haiyun Peng, T. Yang, and E. Cambria. Disentangled variational auto-encoder for semi-supervised learning. *ArXiv*, abs/1709.05047, 2019.

[14] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Advances in Neural Information Processing Systems*, pages 9408–9418, 2019.

[15] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[17] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412, 2019.

[18] Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*, 2019.

[19] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[20] Narayanaswamy Siddharth, Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in neural information processing systems*, pages 5925–5935, 2017.

[21] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[22] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*, 2019.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[24] Matthew Willetts, Stephen Roberts, and Chris Holmes. Disentangling to cluster: Gaussian mixture variational ladder autoencoders. *arXiv preprint arXiv:1909.11501*, 2019.

[25] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[26] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*, 2017.

[27] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12192–12201, 2019.

[28] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020.

6 Appendix

6.1 Upper bound of KL-divergence

$$\begin{aligned}\mathcal{KL}^u\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right) &= \mathcal{KL}(w(\mathbf{y}|\mathbf{x};\eta)\|p(\mathbf{y})) \\ &+ \sum_{k=1}^{K} w(\mathbf{y}=k|\mathbf{x};\eta)\cdot\mathcal{KL}\Big(\mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi),diag\{(\sigma_k(\mathbf{x};\xi))^2\})\|\mathcal{N}(\mathbf{z}|\mu_k^0,diag\{(\sigma_k^0)^2\})\Big).\end{aligned} \tag{14}$$

6.2 Proof of Theorem 1.

Denote the random vector associated with the distribution of the distribution $p(\mathbf{z})$ by $\mathbf{z} = (\mathbf{z}_1,\cdots,\mathbf{z}_d)^\top \sim N(\mu, diag\{\sigma^2\})$ and $q(\mathbf{z}|\mathbf{x})$ by $\mathbf{z}|\mathbf{x} = (\mathbf{z}_1|\mathbf{x},\cdots,\mathbf{z}_d|\mathbf{x})^\top \sim N(\mu(\mathbf{x}), diag\{\sigma(\mathbf{x})^2\})$ (we omitted notations of parameters). Then the expectation of the KL-

divergence between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$ is written as:

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{x}}[\mathcal{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))] \\
= \; & \mathbb{E}_{\mathbf{x}}\Big[\frac{1}{2}\sum_{j=1}^{d}\Big(\frac{[\mathbb{E}(\mathbf{z}_j|\mathbf{x})-\mathbb{E}(\mathbf{z}_j)]^2}{Var(\mathbf{z}_j)} - 1 + \log \mathrm{Var}(\mathbf{z}_j) - \log \mathrm{Var}(\mathbf{z}_j|\mathbf{x}) + \frac{\mathrm{Var}(\mathbf{z}_j|\mathbf{x})}{\mathrm{Var}(\mathbf{z}_j)}\Big)\Big] \\
= \; & \frac{1}{2}\sum_{j=1}^{d}\Big(\frac{\mathbb{E}_{\mathbf{x}}[\mathbb{E}(\mathbf{z}_j|\mathbf{x})-\mathbb{E}_{\mathbf{x}}[\mathbb{E}(\mathbf{z}_j|\mathbf{x})]]^2}{\mathrm{Var}(\mathbf{z}_j)} + \mathbb{E}_{\mathbf{x}}[\log \mathrm{Var}(\mathbf{z}_j)] - \mathbb{E}_{\mathbf{x}}[\log \mathrm{Var}(\mathbf{z}_j|\mathbf{x})] + \frac{\mathbb{E}_{\mathbf{x}}[\mathrm{Var}(\mathbf{z}_j|\mathbf{x})]}{\mathrm{Var}(\mathbf{z}_j)}\Big) - \frac{d}{2} \\
= \; & \frac{1}{2}\sum_{j=1}^{d}\Big(\frac{\mathrm{Var}_{\mathbf{x}}[\mathbb{E}(\mathbf{z}_j|\mathbf{x})]}{\mathrm{Var}(\mathbf{z}_j)} + \frac{\mathbb{E}_{\mathbf{x}}[\mathrm{Var}(\mathbf{z}_j|\mathbf{x})]}{\mathrm{Var}(\mathbf{z}_j)} + \mathbb{E}_{\mathbf{x}}\Big[\log\frac{\mathrm{Var}(\mathbf{z}_j)}{\mathrm{Var}(\mathbf{z}_j|\mathbf{x})}\Big]\Big) - \frac{d}{2} \\
= \; & \frac{1}{2}\sum_{j=1}^{d}\Big(\frac{\mathrm{Var}(\mathbf{z}_j)}{\mathrm{Var}(\mathbf{z}_j)} + \mathbb{E}_{\mathbf{x}}\Big[\log\frac{\mathrm{Var}(\mathbf{z}_j)}{\mathrm{Var}(\mathbf{z}_j|\mathbf{x})}\Big]\Big) - \frac{d}{2} \\
= \; & \frac{1}{2}\sum_{j=1}^{d}\mathbb{E}_{\mathbf{x}}\Big[\log\frac{\mathrm{Var}(\mathbf{z}_j)}{\mathrm{Var}(\mathbf{z}_j|\mathbf{x})}\Big]. \qquad (15)
\end{aligned}
$$

Using above inequality (15), the expectation of second term in (14) is written as:

$$
\begin{aligned}
0 \; \leq \; & \sum_{k=1}^{K}\mathbb{E}_{\mathbf{x}}\left[w(\mathbf{y}=k|\mathbf{x};\eta)\cdot\mathcal{KL}(\mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi), diag\{(\sigma_k(\mathbf{x};\xi))^2\})\|\mathcal{N}(\mathbf{z}|\mu_k^0, diag\{(\sigma_k^0)^2\}))\right] \\
= \; & \sum_{k=1}^{K}\int_{\mathbf{x}} p(\mathbf{x})w(\mathbf{y}=k|\mathbf{x};\eta)\cdot\mathcal{KL}(\mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi), diag\{(\sigma_k(\mathbf{x};\xi))^2\})\|\mathcal{N}(\mathbf{z}|\mu_k^0, diag\{(\sigma_k^0)^2\}))d\mathbf{x} \\
= \; & \sum_{k=1}^{K}\int_{\mathbf{x}} p(\mathbf{y}=k)p(\mathbf{x}|\mathbf{y}=k;\eta)\cdot\mathcal{KL}(\mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi), diag\{(\sigma_k(\mathbf{x};\xi))^2\})\|\mathcal{N}(\mathbf{z}|\mu_k^0, diag\{(\sigma_k^0)^2\}))d\mathbf{x} \\
= \; & \sum_{k=1}^{K} w_k^0\cdot\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}\left[\mathcal{KL}(\mathcal{N}(\mathbf{z}|\mu_k(\mathbf{x};\xi), diag\{(\sigma_k(\mathbf{x};\xi))^2\})\|\mathcal{N}(\mathbf{z}|\mu_k^0, diag\{(\sigma_k^0)^2\}))\right] \\
\leq \; & \frac{1}{2}\sum_{k=1}^{K} w_k^0\sum_{j=1}^{d}\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}\left[\log\frac{(\sigma_{kj}^0)^2}{(\sigma_{kj}(\mathbf{x};\xi))^2}\right],
\end{aligned}
$$

where the KL-divergence is always non-negative and $p(\mathbf{x}|\mathbf{y}=k;\eta) = w(\mathbf{y}=k|\mathbf{x};\eta)p(\mathbf{x})/p(\mathbf{y}=k)$.

Therefore, where the random vector associated with the distribution of the kth component of the mixture distribution $p(\mathbf{z})$ is denoted by $\mathbf{z}^k = (\mathbf{z}_1^k, \cdots, \mathbf{z}_d^k)^\top \sim N(\mu_k^0, diag\{(\sigma_k^0)^2\})$,

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{x}}\mathcal{KL}^u\left(q(\mathbf{z}|\mathbf{x};\eta,\xi)\|p(\mathbf{z})\right) \\
\leq \; & \mathbb{E}_{\mathbf{x}}\mathcal{KL}(w(\mathbf{y}|\mathbf{x};\eta)\|p(\mathbf{y})) + \frac{1}{2}\sum_{k=1}^{K} w_k^0\sum_{j=1}^{d}\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}\left[\log\frac{\mathrm{Var}(\mathbf{z}_j^k)}{\mathrm{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)}\right] \\
\leq \; & \mathbb{E}_{\mathbf{x}}\mathcal{KL}(w(\mathbf{y}|\mathbf{x};\eta)\|p(\mathbf{y})) + \sum_{k=1}^{K}\sum_{j=1}^{d}\frac{w_k^0}{2}\log\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}\left[\frac{\mathrm{Var}(\mathbf{z}_j^k)}{\mathrm{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)}\right]. \qquad (16)
\end{aligned}
$$

This completes the proof.

In addition, from the second term of (16),

$$\begin{aligned}\sum_{k=1}^{K}\sum_{j=1}^{d}\frac{w_k^0}{2}\log\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}\left[\frac{\text{Var}(\mathbf{z}_j^k)}{\text{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)}\right] &= \sum_{k=1}^{K}\sum_{j=1}^{d}\frac{w_k^0}{2}\log\left[\frac{\text{Var}_{\mathbf{x}|\mathbf{y}=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x};\xi)]+\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)]}{\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)]}\right] \\ &= \sum_{k=1}^{K}\sum_{j=1}^{d}\frac{w_k^0}{2}\log\left[1+\frac{\text{Var}_{\mathbf{x}|\mathbf{y}=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x};\xi)]}{\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)]}\right] \\ &\leq \sum_{k=1}^{K}\sum_{j=1}^{d}\frac{w_k^0}{2}\left[\frac{\text{Var}_{\mathbf{x}|\mathbf{y}=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x};\xi)]}{\mathbb{E}_{\mathbf{x}|\mathbf{y}=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x};\xi)]}\right], \end{aligned} \tag{17}$$

by the variance decomposition formula.

In terms of regression, $\text{Var}(\mathbf{z}_j^k)$, $\text{Var}_{\mathbf{x}|y=k}[\mathbb{E}(\mathbf{z}_j^k|\mathbf{x})]$, $\mathbb{E}_{\mathbf{x}|y=k}[\text{Var}(\mathbf{z}_j^k|\mathbf{x})]$ stand for SST (Total Sum of Squares), SSR (Regression Sum of Squares), and SSE (Sum of Squared Errors), respectively. So, the above statistics is closely related to F-statistics: SSR/SSE.

6.3 MNIST dataset

6.3.1 Tuning parameter grid search

First, the KL-divergence measures how the fitted posterior distribution is different from the pre-designed prior distribution and defined by:

$$\text{KL-divergence} = \frac{1}{|I_{test}|}\sum_{i\in I_{test}}\mathcal{KL}^u\left(q(\mathbf{z}|x_i;\eta,\xi)\|p(\mathbf{z})\right). \tag{18}$$

Smaller KL-divergence value means that the fitted posterior distribution structure is more regularized to the pre-designed latent space.

Next, the average negative single-scale structural similarity (negative SSIM) is defined by

$$\text{negativeSSIM}(\mathbf{X}) = \frac{1}{2}\left(1-\frac{1}{|\mathbf{X}|^2}\sum_{(\mathbf{x},\mathbf{x}')\in\mathbf{X}\times\mathbf{X}}\text{SSIM}(\mathbf{x},\mathbf{x}')\right), \tag{19}$$

where $\text{SSIM}(\mathbf{x},\mathbf{x}')$ for $\mathbf{x},\mathbf{x}'\in\mathbf{X}$ is the similarity measure between two images $\mathbf{x}$, $\mathbf{x}'$ and has a value on $[-1,1]$. So, negative SSIM has a value on $[0,1]$ and indicates how many diverse images $\mathbf{X}$ consists of. For $\mathbf{X}$ being the set of images generated from a fitted VAE model, the $\text{negativeSSIM}(\mathbf{X})$ indicates how expressiveness of the fitted VAE model. We consist $\mathbf{X}$ with the images generated from our fitted decoder, where each image is produced from 13×13 equally spaced grid points on the latent space.

Lastly, the classification error is defined by

$$\text{classification error} = \frac{1}{|I_{test}|}\sum_{i\in I_{test}}\mathbb{I}\left(y_i\neq\arg\max_k(w(\mathbf{y}=k|x_i;\eta))\right), \tag{20}$$

where $\mathbb{I}(\cdot)$ is a indicator function and $w(\mathbf{y}|\mathbf{x};\eta)$ is a posterior probability to the pre-designed clusters in our proposed VAE model. The classification error shows the degree of discrepancy between the assigned label by the posterior probability and the true label of data. Thus, the VAE model with low classification error can separate the latent space into subsets on which the labels of data are well identified.

The left panel of Figure 8 indicates that the KL-divergence mostly depends on λ_2 not λ_1. Because λ_2 is the tuning parameter associated with the penalty function preventing β from shrinking toward zero, so a small λ_2 indirectly alleviates the regularization effect of the KL divergence term. In the middle panel of Figure 8, we can observe that the negativeSSIM decreases as λ_2 increases and is almost the same for all λ_1. Since the term measuring the precision of generated samples dominates in the objective function when λ_2 is small, so the fitted VAE model with the small λ_2 is more likely to have higher diversity. The right panel in Figure 8 shows that the classification error mainly depends on λ_1. The classification error decreases as λ_1 increases. Because a large λ_1 restricts the model space of VAE with low negative cross-entropy, the fitted VAE model with the large λ_1 tends to have a low classification error. While, for a fixed λ_1, the classification error does not change much as λ_2 varies.

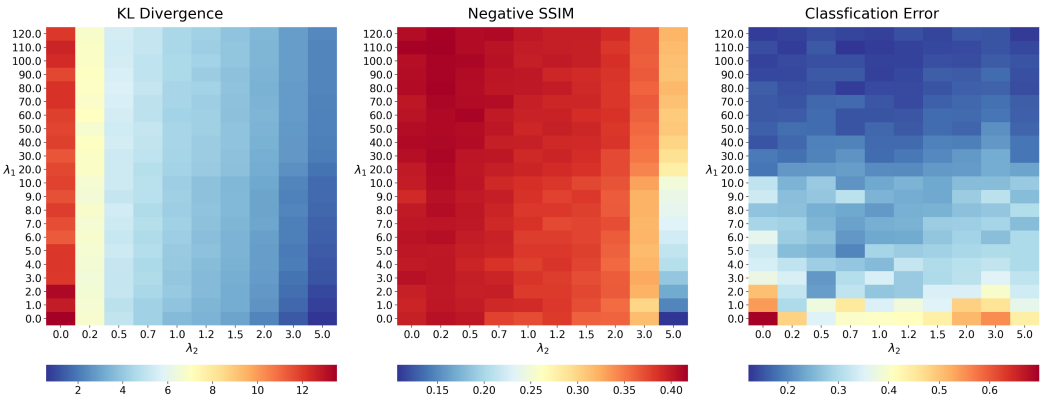

Figure 8: From left to right, heat-maps of KL-divergence, negative SSIM, and the classification error w.r.t. various λ_1 and λ_2.

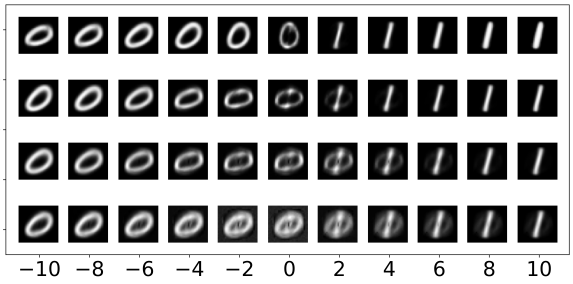

Figure 9: Interpolation produced by various pre-designs of mixture components layouts. From top to the bottom, distances between two component's centers increases as 8, 16, 24, and 32.

6.3.2 Various Pre-designs and Interpolations

We investigate how interpolation results are changed according to various pre-designs of the latent space. We used a small MNIST dataset with only 0 and 1 labels, so a mixture distribution with two Gaussian components is used as a prior distribution. Here is our pre-design: all Gaussian components have diagonal covariance with all diagonal elements 4, and we set their centers as $(-r, 0)^\top, (r, 0)^\top$, respectively. Distance between centers($2r$) was changed as 8, 16, 24, 32, and separate VAE models were fitted for each r. Also, in order to examine the interpolation pattern under the same condition, images were reconstructed in 11 equally spaced line segments from $(-10, 0)^\top$ to $(10, 0)^\top$ on 2-dimensional latent space. Experimental settings that aren't mentioned here are same as Section 4.1's. Figure 9 shows that the variation of interpolation changes slowly if clusters of the prior are placed far from each other.

6.4 CIFAR-10 dataset

6.4.1 Additional interpolation results

Figure 10 interpolation result is given by $(t, D(z(t); \theta)), 0 \leq t \leq 1$ where $z(t) = t \cdot \mu_k(x; \xi) + (1 - t) \cdot \mu_{k'}(x'; \xi)$ for arbitrary different samples x, x' which are belong to different classes k, k', respectively. And Figure 11 and 12 are obtained from $D(z'; \theta)$ where

$$z'_j = \begin{cases} z_j + u \text{ if } j \text{ belongs to the activated latent subspace} \\ z_j \text{ if o.w.} \end{cases}$$

where $j = 1, \cdots, d$ and noise u is sampled from $U(-1, 1)$.

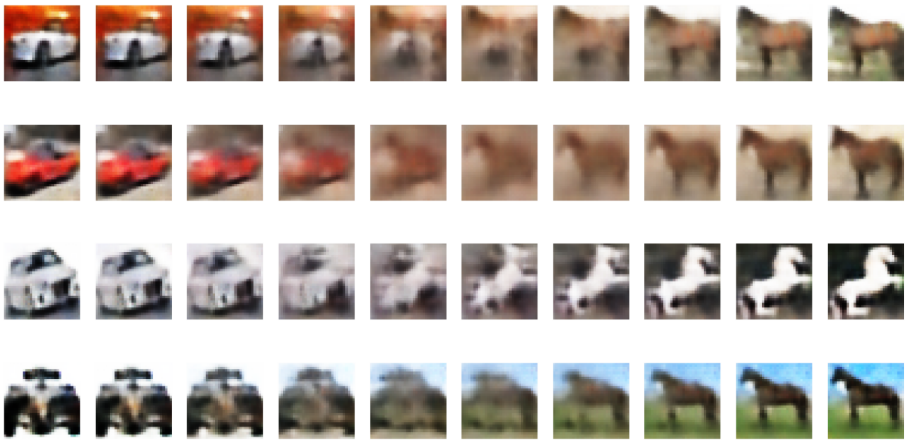

Figure 10: Interpolation between two images belongs to different classes; automobile and horse).

Figure 11: Reconstructed images given latent variables which are added noises on the activated latent subspace. Features of background and objects are changed as the values of activated latent subspace change.

Figure 12: Reconstructed images given latent variables which are added noises on the activated latent subspace. Features of background and objects are changed as the values of activated latent subspace change.

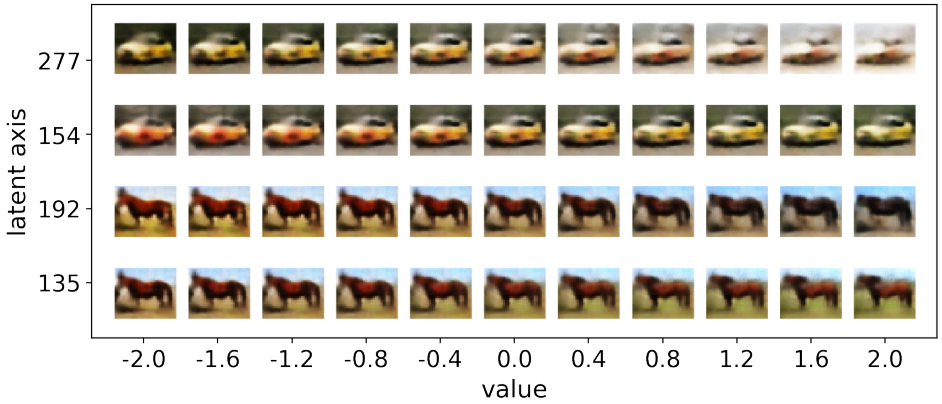

Figure 13: A series of images where only one activated latent axis's value is changed from -2 to 2, and other activated latent subspaces' values are fixed; from top to bottom, used activated latent axes are 277, 154 for automobile class and 192, 135 for horse class. As the value of activated latent subspace is changes, features of generated sample are changed (like the brightness, color of objects, color of backgrounds).

6.4.2 Activated latent subspace measurement

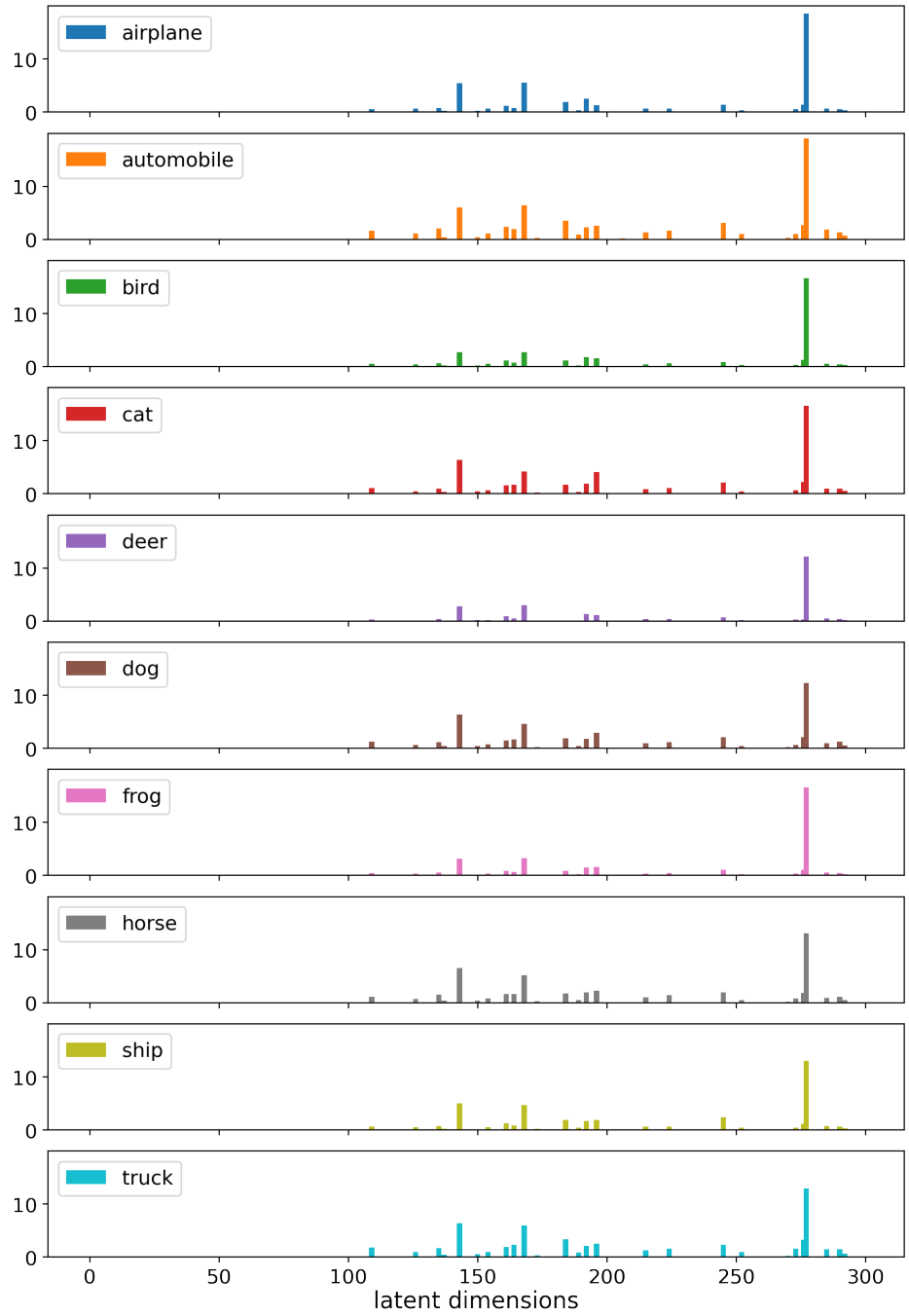

Figure 14: Visualization of the activated latent subspace measurement for each class.

6.4.3 Network architectures used in CIFAR-10 dataset experiment

Table 1: The encoder and decoder network.

Encoder	**Decoder**
$\mathbf{x} \in \mathbb{R}^{32\times32\times3}$	input $\mathbf{z} \in \mathbb{R}^{300}$
5×5 conv, 32 filters, 2 strides, relu, batchnorm	1024 dense, relu, batchnorm
5×5 conv, 64 filters, 2 strides, relu, batchnorm	5×5 convtrans, 256 filters, 2 strides, relu, batchnorm
3×3 conv, 128 filters, 2 strides, relu, batchnorm	5×5 convtrans, 256 filters, 1 strides, relu, batchnorm
3×3 conv, 256 filters, 2 strides, relu, batchnorm	5×5 convtrans, 128 filters, 2 strides, relu, batchnorm
1024 dense, relu, batchnorm	5×5 convtrans, 64 filters, 2 strides, relu, batchnorm
20 × (300 dense, linear)	5×5 convtrans, 32 filters, 2 strides, relu, batchnorm
.	1×1 conv trans, 3 filters, 1 strides, sigmoid

Table 2: The classification network.

Classifier
$\mathbf{x} \in \mathbb{R}^{32\times32\times3}$
3×3 conv, 32 filters, 1 strides, relu
3×3 conv, 32 filters, 1 strides, relu
2×2 maxpooling
3×3 conv, 64 filters, 1 strides, relu
3×3 conv, 64 filters, 1 strides, relu
2×2 maxpooling
3×3 conv, 128 filters, 1 strides, relu
3×3 conv, 128 filters, 1 strides, relu
2×2 maxpooling
128 dense, relu
10 dense, softmax