Relaxed Multivariate Bernoulli Distribution and Its Applications to Deep Generative Models

Xi Wang[*]
School of Data Science and Engineering
East China Normal University
Shanghai 200062, China

Junming Yin
Eller College of Management
University of Arizona
Tucson, AZ 85721

Abstract

Recent advances in variational auto-encoder (VAE) have demonstrated the possibility of approximating the intractable posterior distribution with a variational distribution parameterized by a neural network. To optimize the variational objective of VAE, the reparameterization trick is commonly applied to obtain a low-variance estimator of the gradient. The main idea of the trick is to express the variational distribution as a differentiable function of parameters and a random variable with a fixed distribution. To extend the reparameterization trick to inference involving discrete latent variables, a common approach is to use a continuous relaxation of the categorical distribution as the approximate posterior. However, when applying continuous relaxation to the multivariate cases, multiple variables are typically assumed to be independent, making it suboptimal in applications where modeling dependency is crucial to the overall performance. In this work, we propose a multivariate generalization of the Relaxed Bernoulli distribution, which can be reparameterized and can capture the correlation between variables via a Gaussian copula. We demonstrate its effectiveness in two tasks: density estimation with Bernoulli VAE and semi-supervised multi-label classification.

1 INTRODUCTION

Variational inference (VI) is an optimization-based approach for approximating the intractable posterior distribution of latent variables in complex probabilistic models (Jordan et al., 1999; Wainwright & Jordan, 2008). VI can be scaled to massive data sets using stochastic optimization (Hoffman et al., 2013). With the rise of deep learning, variational auto-encoder (VAE) serves as a bridge between classical variational inference and deep neural networks (Kingma & Welling, 2014; Rezende et al., 2014). The essence of VAE is to employ a neural network to parameterize a function that maps observed variables to the variational parameters of the approximate posterior. One of the most important techniques behind VAE is the reparameterization trick. This trick represents the sampling operation with a differentiable function of parameters and a random variable with a fixed base distribution, which can provide a low-variance gradient estimator for the variational objective function of VAE. From the computation graph perspective, this trick enables one to construct stochastic nodes with random variables (Schulman et al., 2015), in which gradients can propagate from samples to their parameters and parent nodes during the backward computation.

However, when the latent variables are discrete, the reparameterization trick becomes difficult to apply, as a discrete random variable cannot be written as a differentiable transformation of a parameter-independent base distribution. Recently, Maddison et al. (2017) and Jang et al. (2017) propose the Concrete distribution to address this issue. Concrete distribution is a continuous relaxation of the categorical distribution, with which the reparameterization trick can be extended to models involving discrete latent variables. This relaxation technique has been widely used in many applications, including modeling discrete semantic classes (Kingma et al., 2014), learning discrete structures of graph (Franceschi et al., 2019), and neural architecture search (Chang et al., 2019).

It is worth noting that, when this relaxation technique is applied to the multivariate case (e.g., VAE with a multivariate discrete latent space), a common assumption made in practice is to assume independence among all the latent variables (Maddison et al., 2017; Jang et al., 2017). We argue that this approach may not be suitable for certain

[*] Work done while remotely visiting University of Arizona.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

applications. For instance, when performing density estimation with a discrete latent variable model, a factorized posterior would ignore the spatial structure in images. Another example is multi-label learning, where the ground truth label is often represented by a vector of Bernoulli variables. It has been shown that capturing dependencies among different labels can significantly improve the performance (Gibaja & Ventura, 2015).

In this paper, we make an attempt to generalize the Concrete distribution to the multivariate case. We focus on a special case of Concrete distribution: Relaxed Bernoulli. We propose to combine the Gaussian copula and the Relaxed Bernoulli to create a continuous relaxation of the multivariate Bernoulli distribution, which is referred to as RelaxedMVB. It has the following two main advantages: (1) RelaxedMVB can be reparameterized so that sampling from this distribution is differentiable with respect to its parameters; and (2) RelaxedMVB can capture the correlation between multiple Relaxed Bernoulli variables. Our contributions in this work can be summarized as follows:

1. We present RelaxedMVB, a reparameterizable relaxation of the multivariate Bernoulli distribution that explicitly models the correlation structure.
2. We build a Bernoulli VAE with RelaxedMVB as the approximate posterior for density estimation task on the MNIST and Omniglot datasets. We show that incorporating correlation into the variational posterior significantly improves the performance.
3. We generalize the semi-supervised VAE (Kingma et al., 2014) to the multi-label setting using RelaxedMVB. On the CelebA dataset (Liu et al., 2015), we show that: (1) modeling label dependencies can improve classification accuracy; and (2) our model is able to well capture the underlying class structure of the data.

2 RELATED WORK

Multivariate Bernoulli. Several approaches have been proposed to model dependency among Bernoulli variables. Bernoulli mixtures (Bishop, 2006) model multiple binary variables with a mixture of factorized Bernoulli distribution. As such, although the binary variables are independent within each mixture component, they become dependent in the joint distribution. This distribution has been used to capture the correlation between different labels in the multi-label classification problem (Li et al., 2016). Dai et al. (2013) propose the Multivariate Bernoulli distribution, which can model higher order interactions among variables instead of only pairwise interactions. Arithmetic circuits (Darwiche, 2003) and sum-product networks (Poon & Domingos, 2011) use rooted acyclic directed graphs to specify the joint distribution of multiple binary variables. However, these approaches all aim at modeling multivariate Bernoulli in an exact manner. The discontinuous nature of these distributions makes them difficult to be reparameterized and to be integrated into deep generative models.

Copula VI. Tran et al. (2015) use copula to augment the mean-field variational inference for approximating the posterior of continuous latent variables. Neural Gaussian Copula VAE (Wang & Wang, 2019) incorporates the Gaussian copula into VAE in order to address the posterior collapse problem in the continuous latent space. Suh & Choi (2016) adopt the Gaussian copula in the decoder of VAE, which helps to model the dependency structure in observed data. However, none of them can be directly applied to inference involving discrete latent variables.

Structured discrete latent variable models. Constructing latent variable models with structured discrete variables has been discussed in several recent works. For example, Corro & Titov (2019) propose a structured discrete latent variable model for semi-supervised dependency parsing. Yin et al. (2018) introduce StructVAE, a tree-structured discrete latent variable model for semantic parsing. However, all these works aim at building models with specific latent structures for particular applications, while we focus on more general settings. Another example is discrete VAE (Rolfe, 2017). The way that this model accommodates the correlation between discrete latent variables is substantially different from our model: discrete VAE assumes an RBM prior and imposes an autoregressive hierarchy in the approximate posterior of discrete latent variables. Moreover, discrete latent variables in discrete VAE are augmented with a set of auxiliary continuous random variables and the conditional distribution of the observations only depends on the continuous latent space, while the observed variables in our model are directly conditioned on discrete latent variables.

3 BACKGROUND

To provide the necessary background, we begin with a short review of VAE and Relaxed Bernoulli distribution.

3.1 Variational Auto-Encoder (VAE)

Let $\mathbf{x}$ represent observed random variables and $\mathbf{z}$ denote low-dimensional latent variables. The generative model is defined as $p(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$, where $\boldsymbol{\theta}$ is a set of model parameters such as weights and biases of a decoder neural network. Given a training set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the model is trained by maximizing the marginal log-likelihood with respect to $\boldsymbol{\theta}$:

$$\log p(\mathbf{X}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i) = \sum_{i=1}^{N} \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z})p(\mathbf{z})d\mathbf{z}. \tag{1}$$

However, marginalization over the latent variable $\mathbf{z}$ is typically intractable. To sidestep this issue, VAE (Kingma & Welling, 2014) employs a *parametric* variational distribution $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, referred to as an encoder, to approximate the true but intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$. A variational lower-bound, also known as the *evidence lower bound* (ELBO), is then maximized as a surrogate objective instead of directly optimizing the marginal log-likelihood:

$$\begin{aligned}\log p(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})] \\ - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})).\end{aligned} \tag{2}$$

To apply gradient-based optimization methods, one has to estimate the gradient of the first term in the ELBO, i.e., $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})]$. Unbiased gradient with respect to $\boldsymbol{\theta}$ can be easily obtained with a Monte Carlo gradient estimator, but unbiased gradient with respect to $\boldsymbol{\phi}$ is more difficult to compute. A reparameterization trick is commonly applied, which aims to represent the sampling routine $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ as a deterministic and differentiable function $\mathbf{z} = f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$ of an auxiliary random variable $\boldsymbol{\epsilon}$ with a parameter-independent base distribution $q(\boldsymbol{\epsilon})$. In this way, the Monte Carlo estimation of the expectation in Eq. (2) becomes differentiable with respect to $\boldsymbol{\phi}$. More specifically, the gradient can be estimated by:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})] = \mathbb{E}_{q(\boldsymbol{\epsilon})}\left[\frac{\partial}{\partial \boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\mathbf{x} \mid f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}))\right] \\ \approx \frac{1}{M}\sum_{m=1}^{M} \frac{\partial}{\partial \boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\mathbf{x} \mid f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_m, \mathbf{x})),\end{aligned} \tag{3}$$

where $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_M \overset{\text{i.i.d.}}{\sim} q(\boldsymbol{\epsilon})$. In many cases, the encoder $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ is assumed to take a simple form of fully factorized Gaussian, i.e., for a d-dimensional latent variable,

$$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) = \prod_{j=1}^{d} q_{\boldsymbol{\phi}}(z_j \mid \mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})). \tag{4}$$

To reparameterize $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, one only needs to draw a standard d-dimensional Gaussian random vector and then perform an affine transformation. If the prior $p(\mathbf{z})$ is also assumed to be a Gaussian, the KL divergence term in Eq. (2) can be computed analytically. In addition to the Gaussian distribution, the reparameterization trick can also be generalized to distributions in the location-scale family or distributions with a tractable inverse cumulative distribution function (CDF).

3.2 Relaxed Bernoulli Distribution

The reparameterization trick cannot be directly applied to discrete random variables as there is no differentiable function to transform a base distribution to a discrete distribution. Concrete distribution (Maddison et al., 2017; Jang et al., 2017) resolves this issue with a relaxation of the categorical distribution based on the Gumbel-Softmax trick. The binary special case, referred to as the Relaxed Bernoulli (or Binary Concrete) distribution (Maddison et al., 2017, Appendix B), can be considered as a continuous relaxation or approximation of the Bernoulli distribution, with support on the unit interval $(0, 1)$. One key property of the Relaxed Bernoulli distribution is that it can be reparameterized, as the sampling procedure for $B \sim \mathrm{RelaxedBernoulli}(\alpha, \lambda)$ can be described as:

$$\begin{aligned}U &\sim \mathrm{Uniform}(0, 1), \\ L &= \log(\alpha) + \log(U) - \log(1 - U), \\ B &= \frac{1}{1 + \exp(-L/\lambda)},\end{aligned} \tag{5}$$

where $\alpha \in (0, \infty)$ is the location parameter and $\lambda \in (0, \infty)$ is the temperature parameter that controls the degree of approximation. As $\lambda \to 0$, the random variable B converges to Bernoulli with parameter $\alpha/(1 + \alpha)$; as $\lambda \to \infty$, the distribution of B becomes degenerate at 0.5. $\mathrm{RelaxedBernoulli}(\alpha, \lambda)$ can also be directly reparameterized from a logistic random variable $L \sim \mathrm{Logistic}(0, 1)$, followed by an addition of $\log(\alpha)$, a division by λ, and a sigmoid transformation.

4 RELAXED MULTIVARIATE BERNOULLI

Generalizing the Relaxed Bernoulli distribution to the multivariate case is not straightforward, as it is difficult to directly specify its correlation structure in the form of a covariance matrix, in contrast to the multivariate Gaussian or the multivariate t-distribution. As the Relaxed Bernoulli distribution can be reparameterized by applying a deterministic and differentiable transformation to a $\mathrm{Uniform}(0, 1)$ random variable (Eq. (5)), we propose to use the Gaussian copula to characterize the correlation between multiple $\mathrm{Uniform}(0, 1)$ random variables, so that their dependencies can be transferred to multiple Relaxed Bernoulli variables.

A copula (Nelsen, 2007) is a multivariate cumulative distribution function of $(U_1, U_2, ..., U_d)$ over the unit cube $[0, 1]^d$ with uniform marginals, i.e., $U_j \sim \mathrm{Uniform}(0, 1)$ for $j = 1, \ldots, d$. An important member of the copula family is the Gaussian copula, which is constructed from a multivariate Gaussian distribution. Given a correlation matrix $\mathbf{R} \in [-1, 1]^{d\times d}$, the Gaussian copula $C_{\mathbf{R}}$ with parameter $\mathbf{R}$ can be written as

$$C_{\mathbf{R}}(U_1, U_2, ...U_d) = \Phi_{\mathbf{R}}(\Phi^{-1}(U_1), \Phi^{-1}(U_2), \ldots, \Phi^{-1}(U_d)), \tag{6}$$

where $\Phi_{\mathbf{R}}$ stands for the joint CDF of a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance ma-

```
Algorithm 1: Sampling from RelaxedMVB
Input: $d$: dimension of the distribution
       $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_d)$: location vector
       $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$: PSD covariance matrix with $\sigma_j^2$
       as the $j$th diagonal element
       $\lambda$: temperature
Draw a standard normal sample: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
Compute $\mathbf{L} = \mathrm{CholeskyDecomposition}(\boldsymbol{\Sigma})$
Generate a multivariate Gaussian vector: $\boldsymbol{g} = \mathbf{L}\boldsymbol{\epsilon}$
Apply element-wise Gaussian CDF $\Phi_{\sigma_j}$ with mean
  zero and variance $\sigma_j^2$:
      $U_j = \Phi_{\sigma_j}(g_j), \quad j = 1, \dots, d$
Apply inverse CDF of the logistic distribution:
      $l_j = \log(\alpha_j) + \log(U_j) - \log(1 - U_j), \quad j = 1, \dots, d$
Apply the sigmoid function:
      $B_j = \dfrac{1}{1 + \exp(-l_j/\lambda)}, \quad j = 1, \dots, d$
  return $\boldsymbol{B} = (B_1, \dots, B_d) \in (0,1)^d$
```

trix equal to the correlation matrix $\mathbf{R}$, and Φ^{-1} is the inverse CDF of the standard univariate Gaussian distribution. As a consequence, the Gaussian copula allows for generating a vector of correlated random variables $(U_1, U_2, ...U_d)$ on the unit cube with uniformly distributed marginals.

We propose to combine the Gaussian copula and the Relaxed Bernoulli to create a continuous relaxation of the multivariate Bernoulli distribution that allows for inter-dimensional dependence. We name this distribution after **RelaxedMVB**, a relaxation of the multivariate Bernoulli, and it is parameterized by a location vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_d) \in (0, \infty)^d$, a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$, and a temperature $\lambda \in (0, \infty)$. The sampling procedure for $\boldsymbol{B} \in (0,1)^d \sim \mathrm{RelaxedMVB}(\boldsymbol{\alpha}, \boldsymbol{\Sigma}, \lambda)$ is summarized in Algorithm 1. Similar to the Relaxed Bernoulli distribution, sampling from RelaxedMVB is also *differentiable* with respect to its parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$, and the sampling procedure can be interpreted as a *deterministic* transformation of a standard multivariate Gaussian random variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.

In practice, the covariance matrix $\boldsymbol{\Sigma}$ is typically predicted from each observation $\mathbf{x}$ with an encoder network. Additional effort is required to ensure that $\boldsymbol{\Sigma}$ is positive semi-definite (PSD). We propose different parameterization strategies, including low-rank approximation and Cholesky decomposition, for $\boldsymbol{\Sigma}$ in different applications. More details will be discussed in the next section.

5 APPLICATIONS

We demonstrate the application of RelaxedMVB in two tasks: density estimation with Bernoulli VAE and semi-supervised multi-label classification.

5.1 Density Estimation with Bernoulli VAE

In this task, our goal is to learn a VAE with Bernoulli latent variables to fit a distribution for a set of training samples, referred to as density estimation in Maddison et al. (2017). Our generative model and variational posterior distribution are specified as follows:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) &= p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}), \\ p(\mathbf{z}) &= \prod_{j=1}^{d} \mathrm{Bernoulli}(z_j; 0.5), \\ q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) &\approx \mathrm{RelaxedMVB}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}), \lambda). \end{aligned} \tag{7}$$

$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$ is a fully factorized multivariate Bernoulli (for binary data) or Gaussian (for continuous-valued data) whose distribution parameters are outputs of the decoder network. $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ are represented as two separate encoder networks. Both the decoder and encoder networks contain two hidden layers, with 512 and 256 units respectively. Furthermore, the temperature λ is annealed using a similar schedule proposed in Jang et al. (2017): $\lambda = \max(0.5, \exp(-\tau t))$, where t stands for the training step and λ is updated every T steps. In our experiments, we set $T = 100$ and $\tau = 3\mathrm{e}{-5}$.

Notice that directly inferring the full covariance matrix $\boldsymbol{\Sigma}$ would require $d(d+1)/2$ parameters, where d is the dimension of the latent variable $\mathbf{z}$. To reduce the number of parameters, we parameterize $\boldsymbol{\Sigma}$ using a low-rank matrix $\mathbf{V}$ and a vector $\boldsymbol{\sigma^2}$:

$$\begin{aligned} \boldsymbol{\Sigma} &= \mathbf{V}\mathbf{V}^T + \mathrm{diag}(\boldsymbol{\sigma^2}), \\ \mathbf{V} &\in (-1, 1)^{(d,r)}, \boldsymbol{\sigma^2} \in \mathbb{R}_+^d, \end{aligned} \tag{8}$$

where $r \leq d$ is a hyperparameter controlling the rank of $\mathbf{V}$. In this way, $\boldsymbol{\Sigma}$ is guaranteed to be positive definite. We use $\tanh$ and ReLU as the activation functions for $\mathbf{V}$ and $\boldsymbol{\sigma}^2$ respectively so as to ensure that they are within a valid range.

We train our model by optimizing the ELBO in Eq. (2) and compare its performance with a baseline model in which the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ is approximated by a factorized Relaxed Bernoulli (Jang et al., 2017; Maddison et al., 2017). In both models, we choose to approximate the KL divergence term in the ELBO by computing the KL divergence between the discretization of the

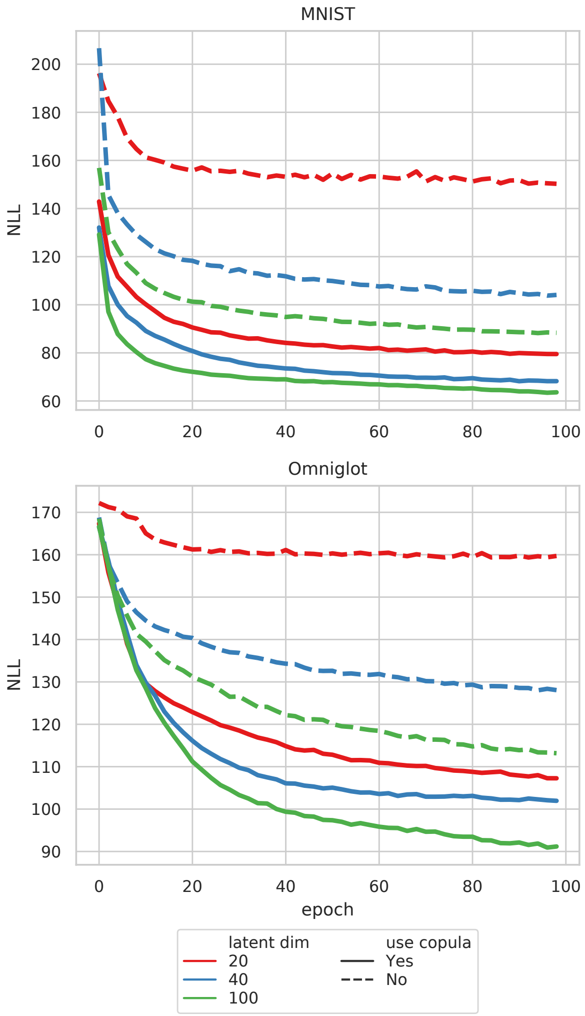


Figure 1: Test loss on the MNIST and Omniglot datasets. The model without copula (dashed line) refers to the baseline in which the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \approx \prod_{j=1}^{d} \mathrm{RelaxedBernoulli}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})_j, \lambda)$.

relaxed posterior and the discrete uniform prior, which corresponds to Eq. (22) in Appendix C of Maddison et al. (2017) and was also used in the official implementation of categorical VAE with the Gumbel-Softmax estimator[1] (Jang et al., 2017).

We conduct the experiments on the MNIST (LeCun et al., 1998) and Omniglot (Lake et al., 2015) datasets. They are datasets of 28×28 images of handwritten digits (MNIST) or letters (Omniglot). For MNIST, we use the standard train/test split; for Omniglot, we use the binarized pre-split version provided by Burda et al. (2015).

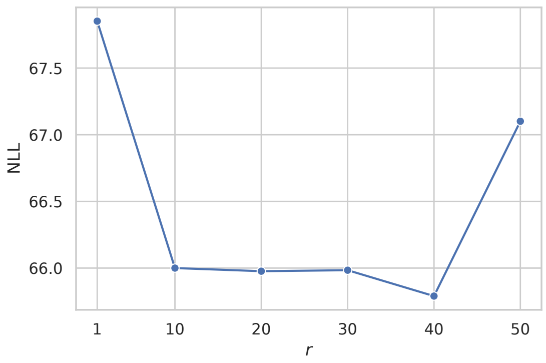


Figure 2: Test loss on MNIST with $d = 100$ as a function of the hyperparameter r.

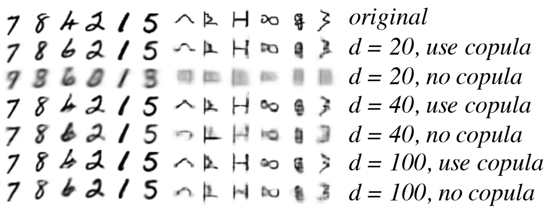


Figure 3: Visualization of the reconstruction result on the test set. Shown in the first row are the original digits and letters. The remaining rows compare the reconstruction quality in different dimensions of the latent space.

We experiment with three different sizes of the latent dimension $d \in \{20, 40, 100\}$. The hyperparameter r is set to be 5, 10, and 20, respectively. Figure 1 shows the test loss in terms of the negative log-likelihood (NLL). It can be observed that our model outperforms the baseline in all the settings and on both datasets, with the most significant improvement achieved at a lower-dimensional latent space ($d = 20$). We also investigate the effect of the hyperparameter r on the MNIST dataset with $d = 100$. As Figure 2 shows, the test loss exhibits a typical U-shaped pattern with the increase of r.

The significant improvement in the test loss can also be reflected in the reconstructed samples on the test set, as shown in Figure 3. By capturing the correlation structure in the latent space, our model is able to reconstruct the original digits and letters with better quality than the baseline without considering correlation. Consistent with the observation in Figure 1, the benefit of modeling inter-dimensional dependencies is more evident when the latent variable is in a lower-dimensional space.

We argue that the inter-dimensional correlation through the covariance matrix of the Gaussian copula helps to capture the spatial structure in the images, which allows our model to learn the distribution of images even with a lower-dimensional latent space. By contrast, the com-

[1] See the fifth code cell in the notebook available at `https://github.com/ericjang/gumbel-softmax/blob/master/Categorical%20VAE.ipynb`.

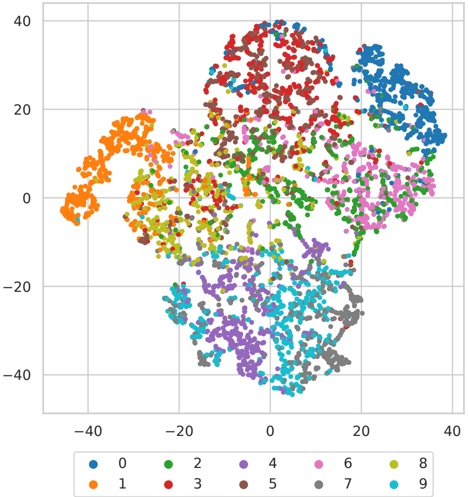


Figure 4: Visualization of the covariance matrix learned from MNIST. We choose the 20-dimensional model as an illustration, i.e., $d = 20, \mathbf{\Sigma} \in \mathbb{R}^{20\times 20}$. Covariance matrices of different digits are highlighted in different colors. It can be observed that the embeddings for digit 7, 4, and 9 are close to each other, as these three digits have similar characteristics.

plex information of the original images cannot be easily captured by a fully factorized binary latent space of very few dimensions. To illustrate the structure learned from the images, we perform t-SNE (van der Maaten & Hinton, 2008) on the upper triangular elements of the covariance matrix $\mathbf{\Sigma}_{\phi}(\mathbf{x})$ learned from MNIST. The embedding shown in Figure 4 demonstrates that our model indeed encodes class-specific and spatial structure information into the learned covariance matrix.

5.2 Semi-supervised Multi-label Classification

Semi-supervised learning involves training a classifier with a small subset of labeled samples and a large subset of unlabeled samples. Kingma et al. (2014) develop a variation of VAE that exploits the power of deep generative models for semi-supervised *multi-class* learning. In this section, we extend this model to the *multi-label* setting, in which each sample can be associated with a subset of all candidate labels.

5.2.1 Semi-supervised VAE

The structure of our model is similar to the generative semi-supervised model (M2) proposed in Kingma et al. (2014). The original model consists of two latent variables: a continuous Gaussian variable $\mathbf{z} \in \mathbb{R}^d$ representing the content information, and a categorical variable y representing the class information, which is observed in the labeled samples. A generative model $p(\mathbf{x}, y, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid y, \mathbf{z})p(y)p(\mathbf{z})$ is trained to: (1) estimate the density of $\mathbf{x}$; and (2) infer unobserved y with a classifier network $q_{\boldsymbol{\phi}}(y \mid \mathbf{x})$.

In the multi-label setting, the class label is represented as a binary vector $\mathbf{y} \in \{0, 1\}^k$, where k is the number of all label candidates. We propose to use our RelaxedMVB to approximate $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ for $\mathbf{y} \in \{0, 1\}^k$, which enables us to capture the correlation between different label candidates and to backpropagate directly with a single sample from $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$. The generative semi-supervised model and the variational posterior are specified as follows:

$$\begin{aligned}
p(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})p(\mathbf{y})p(\mathbf{z}),\\
p(\mathbf{y}) &= \prod_{j=1}^{k} \mathrm{Bernoulli}(y_j; 0.5),\\
p(\mathbf{z}) &= \mathcal{N}(\mathbf{0}, \mathbf{I}_d),\\
q_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{z} \mid \mathbf{x}) &= q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{y}, \mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x}),\\
q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x}) &\approx \mathrm{RelaxedMVB}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}), \lambda),\\
q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{y}, \mathbf{x}) &= \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{x}))).
\end{aligned} \tag{9}$$

$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ is a multivariate Gaussian distribution with the mean vector output by the decoder network and an identity covariance matrix. The ELBO for an unlabeled sample $\mathbf{x}$ is

$$\begin{aligned}
\mathcal{U}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{y},\mathbf{z}\mid\mathbf{x})}\Big[&\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) + \log p(\mathbf{y}) +\\
&\log p(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{y}, \mathbf{z} \mid \mathbf{x})\Big].
\end{aligned} \tag{10}$$

Computing this ELBO and its gradient requires taking expectation with respect to $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$. Kingma et al. (2014) compute the expectation by summing over all possible values of $\mathbf{y}$, which is impractical in the multi-label setting because the computational complexity scales exponentially with the number of label candidates k. However, with RelaxedMVB that is reparameterizable, both the ELBO and its gradient can be efficiently estimated by drawing only a single sample from $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$, which significantly reduces the computational cost.

For a sample $\mathbf{x}$ with observed label $\mathbf{y}$, the ELBO is

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}\mid\mathbf{x},\mathbf{y})}\Big[&\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) + \log p(\mathbf{y}) +\\
&\log p(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}, \mathbf{y})\Big].
\end{aligned} \tag{11}$$

It is worth noting that $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ contributes only to the ELBO in Eq. (10) for unlabeled samples, so labeled samples are completely ignored when training this classifier network. As a solution, Kingma et al. (2014) propose to add a discriminative term $\mathbb{E}_{\tilde{p}_l}(\mathbf{x},\mathbf{y})[\log q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})]$ to the ELBO in Eq. (11), where $\tilde{p}_l(\mathbf{x},\mathbf{y})$ is the empirical distribution of labeled samples. The final objective function to be maximized can be written as:

$$\mathcal{J} = \sum_{(\mathbf{x},\mathbf{y})\sim\tilde{p}_l} \mathcal{L}(\mathbf{x},\mathbf{y}) + c \cdot \sum_{(\mathbf{x},\mathbf{y})\sim\tilde{p}_l} \log q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x}) + \sum_{\mathbf{x}\sim\tilde{p}_u} \mathcal{U}(\mathbf{x}), \tag{12}$$

where $\tilde{p}_u$ is the empirical distribution of unlabeled samples and c is a hyperparameter controlling the relative weight of the discriminative term.

5.2.2 Discriminative Objective

The discriminative term $\mathbb{E}_{\tilde{p}_l(\mathbf{x},\mathbf{y})}[\log q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})]$ in Eq. (12) plays a very import role in semi-supervised VAE. In Kingma et al. (2014) where $q_{\boldsymbol{\phi}}(y \mid \mathbf{x}) \sim \mathrm{Categorical}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x}))$, maximizing this term is equivalent to training a probabilistic classifier whose conditional density function $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ is parameterized by $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$, the output of a network on the labeled samples. However, in our case, $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x}) \approx \mathrm{RelaxedMVB}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}), \lambda)$ is a continuous relaxation and has support on $(0,1)^k$, as we would like sampling from $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ to be differentiable with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$. As a consequence, the likelihood becomes zero at observed labels $\mathbf{y} \in \{0,1\}^k$. A common choice for addressing this issue in the Relaxed Categorical or Relaxed Bernoulli case is to minimize the cross-entropy loss[2] between the predicted logits $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$ and the ground truth label $\mathbf{y}$. However, applying this technique to our case only involves updating the parameters of the encoder network for $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$. As a result, the encoder network for $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ would be completely ignored during training.

To address this problem, we propose a sampling-based training procedure that takes both networks for $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ into account. Our goal is to train $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$ so that samples generated from $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$ are close to the observed labels measured by the L_2 distance. Sampling from $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$ is straightforward: we first feed input data $\mathbf{x}$ into the networks for $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$, and next we generate a sample $\hat{\mathbf{y}}$ from $\mathrm{RelaxedMVB}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x}), \lambda)$ using Algorithm 1. The L_2 distance between $\hat{\mathbf{y}}$ and observed label $\mathbf{y}$ is then minimized with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$. Since $\hat{\mathbf{y}}$ is a differentiable and deterministic function of $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$, the gradients of L_2 distance can be backpropagated to both $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$ (i.e., the reparameterization trick). As a result, parameters of both networks $\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ get updated during backpropagation.

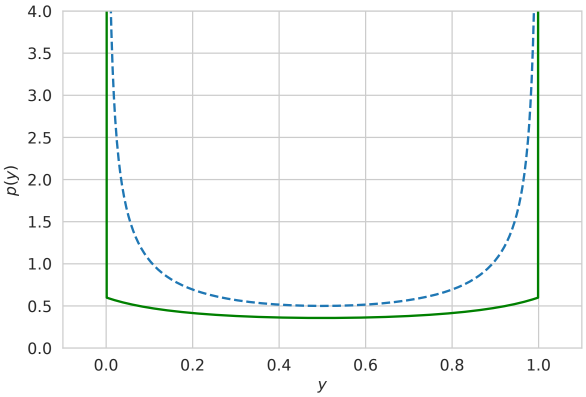


Figure 5: Green solid line: density of the Hard Concrete distribution transformed from $\mathrm{RelaxedBernoulli}(\alpha = 1, \lambda = 0.5)$ with $\zeta = -0.2$ and $\gamma = 1.2$. Blue dashed line: density of $\mathrm{RelaxedBernoulli}(\alpha = 1, \lambda = 0.5)$.

Recall that samples from $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$ are in $(0,1)^k$ because of the relaxation, while the observed labels are in $\{0,1\}^k$. As a result, generated samples close to 0 or 1 but not exactly binary would incur a loss. For example, given the observed label $\mathbf{y} = [1,0]$ and sampled $\hat{\mathbf{y}} = [0.9, 0.1]$ from $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x})$, the L_2 loss in this case could be unnecessary as it may be caused by the continuous nature of $\mathrm{RelaxedMVB}$ instead of poorly learned $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}$. In order to have a better measure of the distance between generated samples and observed labels, we propose to apply a differentiable transformation on each generated sample $\hat{\mathbf{y}}$ so that the resulting $\tilde{\mathbf{y}}$ becomes closer to binary. Here, we adopt the idea of the Hard Concrete distribution proposed in Louizos et al. (2018): given a sample $\hat{\mathbf{y}} \in (0,1)^k$ from RelaxedMVB, we first stretch each dimension of $\hat{\mathbf{y}}$ into a larger interval (γ, ζ), then we apply a hard-sigmoid on it to clip it back into $[0,1]^k$:

$$\bar{\mathbf{y}}_j = \hat{\mathbf{y}}_j(\zeta - \gamma) + \gamma, \qquad \tilde{\mathbf{y}}_j = \min(1, \max(0, \bar{\mathbf{y}}_j)). \tag{13}$$

As Figure 5 illustrates, the probability density of $\tilde{\mathbf{y}}$ is now more concentrated at 0 and 1. As we will show in Section 5.2.4, applying this differentiable transformation is crucial for the overall performance.

5.2.3 Experimental Setup

We test our model on the CelebA dataset of celebrity images (Liu et al., 2015). Each image can be associated with multiple facial attributes, such as smiling and wearing eyeglasses. We randomly select $80,000$ images from the dataset and crop them to the size of 64×64. We then manually choose 25 attributes out of the original 40 attributes to perform semi-supervised multi-label classification. A different subset of $2,000$ images are used as the test set.

The encoder network is composed of a three-layer convolution neural network (CNN) followed by a linear layer

[2] See the semi-supervised VAE tutorial from the Deep Bayes summer school: https://github.com/bayesgroup/deepbayes-2019/tree/master/seminars/day2

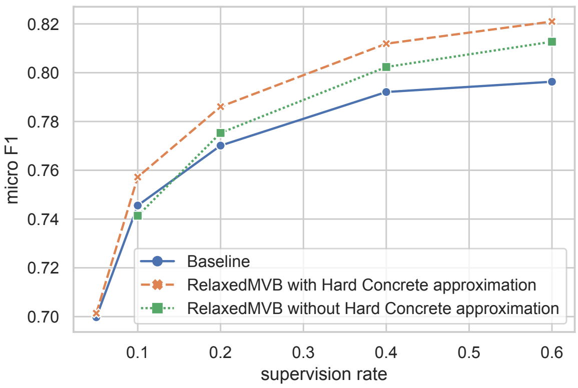


Figure 6: Micro-F1 score on the CelebA dataset under different supervision rates. The baseline refers to a similar model defined in Eq. (9), except that $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x}) \approx \prod_{j=1}^{k} \text{RelaxedBernoulli}(\boldsymbol{\alpha}_{\boldsymbol{\phi}}(\mathbf{x})_j, \lambda)$.

with 256 hidden units. The decoder network is made up of two linear layers with 256 hidden units and a three-layer deconvolution network. We use three separate encoder networks for $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\Sigma}$. As for the parameterization of the covariance matrix $\boldsymbol{\Sigma}$, we choose to let the encoder network directly output its Cholesky factor[3] $\mathbf{L}$, i.e., $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$.

Models are trained with Adam (Kingma & Ba, 2015) for a maximum of 80 epochs and are early stopped if test accuracy does not decrease for 8 consecutive epochs. The initial learning rate is $5\text{e}{-4}$ and is decayed by a factor of 0.999 every epoch. The temperature λ is annealed according to $\lambda = \max(0.5, \lambda_0 \tau^t)$, where λ_0 represents the initial temperature. τ and t stand for the annealing rate and the epoch respectively. We initialize $\lambda_0 = 1$ and set $\tau = 0.99$. The hyperparameter c in the discriminative term of Eq. (12) is set to be 512. The dimension of $\mathbf{z}$ is chosen to be $d = 32$ for all the experiments.

5.2.4 Classification Result

We use a variant of semi-superverised VAE defined in Eq. (9) as the baseline model, in which the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ is instead approximated by a factorized Relaxed Bernoulli (Jang et al., 2017; Maddison et al., 2017).

We compare the classification accuracy of our model with the baseline under the supervision rate ranging from 0.05 to 0.6. We also evaluate a variant of our model that does not apply the differentiable transformation (13). The accuracy is evaluated by the micro-F1 score. As shown in Figure 6, our model outperforms the baseline across all the ranges of the supervision rate, with the most significant improvement occurring in the regime of higher supervision rate. We believe this is because when the number of labeled samples is small, the parameters of the encoder network $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{x})$ cannot be well estimated with a limited amount of labeled samples. Figure 6 also shows that the differentiable transformation in Eq. (13) is essential for achieving better prediction performance.

5.2.5 Inferring the Correlation of Attributes

Since we model inter-attribute dependencies via the Gaussian copula, we can use our trained model to infer the correlation matrix between different attributes on the test set. The procedure is as follows: we first discretize the sample generated from $q_{\boldsymbol{\phi}}(\mathbf{y} \mid \mathbf{x})$ for each $\mathbf{x}$ in the test set and then we compute the empirical correlation matrix on all the samples. This inferred correlation matrix is then compared with the empirical correlation matrix computed on the ground truth labels of the test set. With 20 percent of the labeled samples, our inferred correlation matrix is able to have 281 out of 300 attribute pairs with a correct sign. Furthermore, the average L_2 distance between the two matrices is 0.0017. As an illustration, we plot a subset of both correlation matrices in Figure 7.

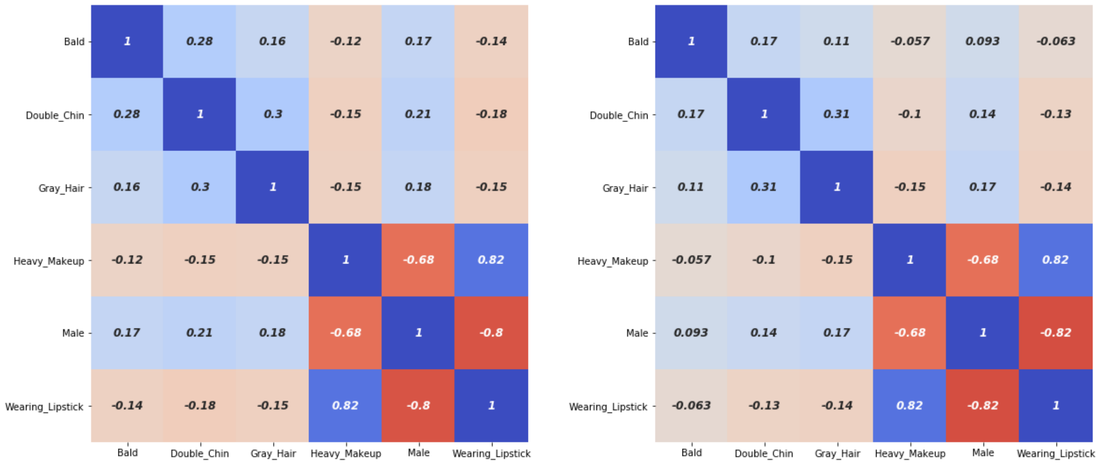
5.2.6 Conditional Generation

Recall that our generative model is specified as $p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) p(\mathbf{y}) p(\mathbf{z})$. To generate a new face image, we fix $\mathbf{y}$ to be a binary vector that represents a set of desired facial attributes, then we sample a continuous variable $\mathbf{z}$ from the prior $p(\mathbf{z})$, and finally pass them to the learned decoder $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{y}, \mathbf{z})$ to generate an observation $\mathbf{x}$. Figure 8 shows generated faces for a set of selected attribute combinations, demonstrating that the decoder can well capture the underlying class structure of the data.

6 CONCLUSION

We present RelaxedMVB, a relaxation of the multivariate Bernoulli distribution that supports reparameterization. The proposed distribution employs a Gaussian copula to allow inter-dimensional correlation to be captured. When RelaxedMVB is integrated into variational auto-encoder, the resulting models show superior performance in two tasks: density estimation and semi-supervised multi-label classification. In future work, we plan to explore more applications of RelaxedMVB. Moreover, it would be interesting to see if our approach can be combined with other gradient estimators, such as RELAX (Grathwohl et al., 2018) and direct optimization through $\arg\max$ (Lorberbom et al., 2019).

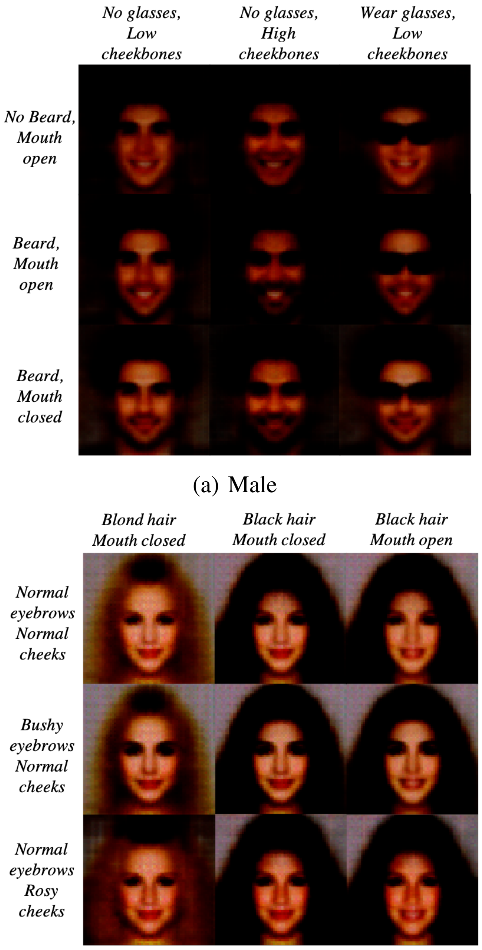
[3] We choose to parameterize the covariance matrix with its Cholesky factor instead of a low-rank approximation because the low-rank approach does not reduce the number of parameters significantly in this scenario, where $k = 25$. When the number of attributes becomes larger, one should consider using low-rank approximation as in Eq. (8).



(a) Empirical correlation matrix.

(b) Inferred correlation matrix.

Figure 7: Comparison between the empirical correlation matrix computed on the ground truth labels of the test set and the correlation matrix inferred by our model.


No glasses, Low cheekbones
No glasses, High cheekbones
Wear glasses, Low cheekbones
No Beard, Mouth open
Beard, Mouth open
Beard, Mouth closed
Blond hair Mouth closed
Black hair Mouth closed
Black hair Mouth open
Normal eyebrows Normal cheeks
Bushy eyebrows Normal cheeks
Normal eyebrows Rosy cheeks

(a) Male

(b) Female

Figure 8: Illustration of conditional generation.

Acknowledgements

We thank the reviewers for their constructive feedback. This work is partially supported by Amazon AWS Machine Learning Research Award (JY).

References

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2015.

Chang, J., Zhang, X., Guo, Y., Meng, G., Xiang, S., and Pan, C. Differentiable architecture search with ensemble Gumbel-Softmax. *arXiv preprint arXiv:1905.01786*, 2019.

Corro, C. and Titov, I. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. In *International Conference on Learning Representations*, 2019.

Dai, B., Ding, S., and Wahba, G. Multivariate Bernoulli distribution. *Bernoulli*, 19(4):1465–1483, 2013.

Darwiche, A. A differential approach to inference in bayesian networks. *Journal of the ACM*, 50(3):280–305, 2003.

Franceschi, L., Niepert, M., Pontil, M., and He, X. Learning discrete structures for graph neural networks. In *Proceedings of the International Conference on Ma-*

chine Learning, pp. 1972–1982, 2019.

Gibaja, E. and Ventura, S. A tutorial on multilabel learning. *ACM Computing Surveys*, 47(3):1–38, 2015.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, C., Wang, B., Pavlu, V., and Aslam, J. Conditional Bernoulli mixtures for multi-label classification. In *Proceedings of the International Conference on Machine Learning*, pp. 2482–2491, 2016.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Lorberbom, G., Gane, A., Jaakkola, T., and Hazan, T. Direct optimization through $\arg\max$ for discrete variational auto-encoder. In *Advances in Neural Information Processing Systems*, pp. 6200–6211, 2019.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through L_0 regularization. In *International Conference on Learning Representations*, 2018.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Nelsen, R. B. *An Introduction to Copulas*. Springer, 2007.

Poon, H. and Domingos, P. Sum-product networks: A new deep architecture. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 337–346, 2011.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning*, pp. 1278–1286, 2014.

Rolfe, J. T. Discrete variational autoencoders. In *International Conference on Learning Representations*, 2017.

Schulman, J., Heess, N., Weber, T., and Abbeel, P. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pp. 3528–3536, 2015.

Suh, S. and Choi, S. Gaussian copula variational autoencoders for mixed data. *arXiv preprint arXiv:1604.04960*, 2016.

Tran, D., Blei, D., and Airoldi, E. M. Copula variational inference. In *Advances in Neural Information Processing Systems*, pp. 3564–3572, 2015.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008.

Wang, P. Z. and Wang, W. Y. Neural Gaussian copula for variational autoencoder. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Yin, P., Zhou, C., He, J., and Neubig, G. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 754–765, 2018.