Learning Discrete and Continuous Factors of Data via Alternating Disentanglement

Yeonwoo Jeong [1] **Hyun Oh Song** [1]

Abstract

We address the problem of unsupervised disentanglement of discrete and continuous explanatory factors of data. We first show a simple procedure for minimizing the total correlation of the continuous latent variables without having to use a discriminator network or perform importance sampling, via cascading the information flow in the β-vae framework. Furthermore, we propose a method which avoids offloading the entire burden of jointly modeling the continuous and discrete factors to the variational encoder by employing a separate discrete inference procedure.

This leads to an interesting alternating minimization problem which switches between finding the most likely discrete configuration given the continuous factors and updating the variational encoder based on the computed discrete factors. Experiments show that the proposed method clearly disentangles discrete factors and significantly outperforms current disentanglement methods based on the disentanglement score and inference network classification score. The source code is available at https://github.com/snu-mllab/DisentanglementICML19.

1. Introduction

Learning to disentangle the underlying explanatory factors of data without supervision is a crucial task for representation learning in AI-related tasks such as speech, object recognition, natural language processing, and transfer learning (Bengio et al., 2013). While establishing a clear quantifiable objective is difficult, in a successfully disentangled representation, a single latent unit of the representation should correspond to a change in a single generative factor of the data while being relatively invariant to others.

To this end, a line of research on unsupervised disentanglement has been pursued under the Variational Autoencoder framework (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018; Gao et al., 2018). The common theme among the recently proposed methods is to penalize the *total correlation* among the latent variables so that the model is encouraged to learn statistically independent factors of data.

Although penalizing the total correlation is important we argue that this alone is not sufficient for learning disentangled representations. Most recent methods focus on learning only the continuous factors of variation and jointly modeling both the continuous and discrete factors of variation is relatively much less studied. When modeling complex and high-dimensional data such as raw images, it becomes difficult to disentangle the discrete factors of data (*i.e.* number of light sources, categorical shape of present objects) from continuous factors (*i.e.* translation, rotation, color) under these methods. Makhzani et al. (2015) have demonstrated that providing the true discrete factors of data to the autoencoder via supervision drastically improves the quality of the learned continuous factors compared to the purely unsupervised case. We hypothesize that lumping both the continuous and discrete factors into a single latent vector and optimizing for the joint variational posterior severely deteriorates the disentanglement performance as this imposes too much modeling burden to the posterior.

In this paper, we first propose a simple procedure for penalizing the total correlation which does not require any extra discriminator network or having to run expensive importance sampling in the β-VAE framework. Then, we propose an alternating disentanglement method where it alternates between finding the most likely configuration of the discrete factors given the continuous factors and updating the inference parameters given the discrete configuration. The empirical results show that decoupling the disentanglement process for continuous and discrete factors via the proposed alternating method leads to strong disentanglement performance both qualitatively and quantitatively.

Our quantitative results on 1) dSprites (Matthey et al., 2017)

[1]Department of Computer Science and Engineering, Seoul National University, Seoul, Korea. Correspondence to: Hyun Oh Song <hyunoh@snu.ac.kr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

dataset on the disentanglement evaluation metric by (Kim & Mnih, 2018), and on 2) the inference network classification score on the learned discrete factors show state of the art results outperforming recently proposed disentanglement methods: $\beta-$VAE, AnchorVAE, FactorVAE, and JointVAE by a large margin.

2. β-VAE and disentanglement

We first review and analyze how the β-VAE framework relates to disentanglement from information theoretic perspective. VAE is a latent variable model that pairs a top-down decoding generator (θ) and a bottom-up encoding inference network (ϕ). Then, a variational lower bound of the marginal log-likelihood, $\mathbb{E}_{x\sim p(x)} \log p(x)$, is maximized. Concretely, the VAE objective is,

$$\begin{aligned}\mathcal{L}(\theta,\phi) = \mathbb{E}_{x\sim p(x)}\big[&\mathbb{E}_{z\sim q_\phi(\cdot|x)} \log p_\theta(x \mid z) \\ &- \beta D_{\mathrm{KL}}\left(q_\phi(z \mid x) \parallel p(z)\right)\big],\end{aligned} \tag{1}$$

where $p(z)$ is the fully factorized standard normal prior. Note, maximizing the objective in Equation (1) can be viewed as maximizing the lower bound on the mutual information between the data x and the latent code z with the KL term. Concretely,

$$\begin{aligned}&I(x;z) - \beta\mathbb{E}_{x\sim p(x)} D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) \\ &= H(x) + \int q_\phi(x,z) \log q_\phi(x \mid z) dz dx \\ &\quad - \beta\mathbb{E}_{x\sim p(x)} D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) \\ &\geq H(x) + \mathbb{E}_{x\sim p(x)}\big[\mathbb{E}_{z\sim q_\phi(\cdot|x)} \log p_\theta(x \mid z) \\ &\quad - \beta D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z))\big],\end{aligned} \tag{2}$$

where the data entropy can be ignored as there is no dependence on the parameters. Also, the KL term in Equations (1) and (2) can be factorized as below (Kim & Mnih, 2018; Chen et al., 2018),

$$\begin{aligned}&\mathbb{E}_{x\sim p(x)} D_{\mathrm{KL}}(q(z|x) \,||\, p(z)) = I(x;z) + D_{\mathrm{KL}}(q(z)||p(z)) \\ &= I(x;z) + \underbrace{D_{\mathrm{KL}}(q(z) \parallel \prod_j q(z_j))}_{=\text{ Total correlation, } TC(z)} + \sum_j D_{\mathrm{KL}}(q(z_j) \parallel p(z_j)),\end{aligned} \tag{3}$$

where $q(z)$ denotes the marginal posterior computed as $q(z) = \mathbb{E}_{x\sim p(x)} q(z \mid x)$. The second term in the factorization is known as *total correlation* (TC) and is a popular measure quantifying the redundancy among a set of m random variables (Watanabe, 1960). The significance of TC is that penalizing TC causes the model to learn statistically independent factors in the data which is a crucial component in disentangled representations (Bengio et al., 2013).

3. Total correlation penalization with information cascading

β-VAE indirectly penalizes $TC(z)$ by increasing the β coefficient to a high value in the KL divergence term in Equation (1) (Higgins et al., 2017). However, this inevitably penalizes the $I(x;z)$ term altogether (see Equation (3)) leading to reduction in the amount of information in z about x. To address this, FactorVAE decreases β and introduces an additional regularization term between the marginal variational posterior and factorized marginal $D_{\mathrm{KL}}(q(z) \parallel \prod_i^m q(z_i))$ using the density-ratio trick via a separate discriminator network (Kim & Mnih, 2018). β-TCVAE on the otherhand estimates TC via importance sampling within minibatches (Chen et al., 2018). We first make the following observations and show a simple practical procedure for minimizing TC without having to rely on additional neural networks or sampling procedures in the β-VAE framework.

Proposition 1. *The mutual information between a single random variable and the rest can be factorized as*

$$I(z_{1:i-1}; z_i) = TC(z_{1:i}) - TC(z_{1:i-1})$$

Proof. See supplementary A1. □

Proposition 2. *The mutual information between x and partitions of $z = [z_1, z_2]$ can be factorized as,*

$$I(x; [z_1, z_2]) = I(x; z_1) + I(x; z_2) - I(z_1; z_2)$$

Proof. See supplementary A2. □

Now, by telescoping sum, we can write,

$$\begin{aligned}TC(z) &= \underbrace{TC(z_{1:2})}_{=I(z_1;z_2)} + \sum_{i=3}^m \left(\underbrace{TC(z_{1:i}) - TC(z_{1:i-1})}_{=I(z_{1:i-1};z_i)}\right) \\ &= \sum_{i=2}^m I(z_{1:i-1}; z_i)\end{aligned} \tag{4}$$

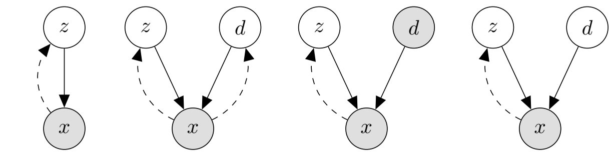
This is because the first term in Equation (4) is equal to $I(z_1; z_2)$ by definition of mutual information and the rest of the terms are equal to $I(z_{1:i-1}; z_i)$ by Proposition 1. Now we aim at penalizing $TC(z)$ by sequentially penalizing the individual summand in Equation (4). From Proposition 2, we can write

$$I(x; z_{1:i}) = I(x; z_{1:i-1}) + I(x; z_i) - I(z_{1:i-1}; z_i)$$

This factorization motivates a maximization algorithm sequentially updating the left hand side $I(x; z_{1:i})$ for all $i = 2, \ldots, m$ which in turn minimizes each summand in Equation (4). Also, from the lower bound of mutual information in Equation (2) we have,

$$I(x; z_{1:i}) \geq H(x) + \mathbb{E}_x \mathbb{E}_{z_{1:i}\sim q_\phi(\cdot|x)} \log p_\theta(x \mid z_{1:i})$$

We maximize $I(x; z_{1:i})$ by maximizing its lower bound $\mathbb{E}_x \mathbb{E}_{z_{1:i}\sim q_\phi(\cdot|x)} p_\theta(x|z_{1:i})$. In practice, we observed it is sufficient to maximize the objective in Equation (1) while penalizing $z_{i+1:m}$ with a large beta coefficient on



(a) β-VAE (b) JointVAE (c) AAE-S (d) Ours

Figure 1. Graphical models view of β-VAE, JointVAE, AAE with supervised discrete variables, and our method. Solid lines denote the generative process and the dashed lines denote the inference process. x, z, d denotes the data, continuous latent code, and the discrete latent code respectively.

$D_{\mathrm{KL}}(q_\phi(z_{i+1:m} \mid x) \parallel p(z_{i+1:m}))$[1]. This leads to sequential updates where each latent dimensions are heavily penalized with high β in the beginning but are sequentially relieved one at a time with small β in a cascading fashion.

Since this procedure implicitly penalizes the individual summand $I(z_{1:i-1}, z_i)$ in the factorization of TC, we can interpret that the method separately controls the information flow one latent variable at a time encouraging the model to encode one statistically independent factor of information per each newly *opened* variable.

To contrast the procedure with related methods, Burgess et al. (2018) and JointVAE increases the KL capacity term which is *shared* across all latent variables. AnchorVAE sets β to a small value for a subset of variables and high value for the rest of the variables but does not dynamically control the information flow (Gao et al., 2018).

4. Alternating disentanglement of discrete and continuous factors

Figure 1 illustrates the graphical models view of the generative and inference processes for β-VAE (Higgins et al., 2017), JointVAE (Dupont, 2018), AAE with supervised discrete factors (Makhzani et al., 2015), and our proposed model respectively. As illustrated in Figure 1b, JointVAE can be viewed as augmenting the continuous latent variables with discrete latent variables ($z = [z'; d]$) in the β-VAE framework. However, simply lumping the latent variables together offloads the entire burden of jointly modeling both the continuous and discrete factors to the variational posterior which can be very challenging.

Makhzani et al. (2015) have investigated learning continuous latent representations while providing the discrete factors through supervision (*i.e.* providing class labels in MNIST) and demonstrated that the model can learn drastically better continuous representations when the burden of simultaneously modeling the continuous and discrete factors is relieved. Inspired by these findings, our idea is to alternate between finding the most likely discrete configuration of the variables given the continuous factors and updating the parameters (ϕ, θ) given the discrete configurations.

Figure 1c and Figure 1d illustrate that the generative and inference process between the two methods resemble in the sense that they only encode the continuous factor z, while Figure 1b encodes both the continuous and discrete factors z, d simultaneously. Unlike AAE-S, discrete factors are not observed in our case.

The joint distributions $q(x, z, d)$ of JointVAE and AAE-S are

$$q_\phi(x, z, d) = p(x) q_\phi(z, d|x)$$

and

$$q_\phi(x, z, d) = \begin{cases} q_\phi(x, z) & \text{if } d = y \\ 0 & \text{otherwise} \end{cases},$$

where y is the provided ground truth discrete factors, respectively. We define the joint distribution similar to AAE-S as

$$q_{\phi,\theta}(x, z, d) = \begin{cases} q_\phi(x, z) & \text{if } d = \operatorname{argmax}_{\hat{d}} p_\theta(x \mid z, \hat{d}) \\ 0 & \text{otherwise} \end{cases},$$

where $q_\phi(x, z) = p(x) q_\phi(z \mid x)$. Likewise, the variational posterior is defined as,

$$q_{\theta,\phi}(z, d \mid x) = \begin{cases} q_\phi(z \mid x) & \text{if } d = \operatorname{argmax}_{\hat{d}} p_\theta(x \mid z, \hat{d}) \\ 0 & \text{otherwise} \end{cases}.$$

Note in our case the inference for the discrete factors involves both the encoder (ϕ) and the decoder (θ) in contrast to JointVAE.

Note from the factorization in Equation (3), the KL term in β-VAE is factorized as the sum of the mutual information term and the divergence from the marginal posterior to the prior. Since the mutual information for discrete variables, $I(x, d)$ is bounded above by $H(d)$, we only need to consider the prior divergence term. For discrete uniform prior, the following lemma shows a useful upperbound which is easier to optimize directly[2].

Lemma 1. *If $p(d_i)$ is the discrete uniform distribution supported on a finite set of cardinality S_i, $D_{KL}(q(d)||p(d)) \leq S\mathbb{E}_{d,d' \sim q(d)}[\mathbb{1}(d = d')] - 1$ where $S = \prod_i S_i$.*

Proof. Denote $p(d) = \prod_i p(d_i)$ which is also discrete uniform.

$$D_{KL}(q(d)||p(d)) = \sum_d q(d) \log \frac{q(d)}{p(d)} \leq \sum_d q(d) \left(\frac{q(d)}{p(d)} - 1 \right)$$
$$= S \sum_d q^2(d) - 1 = S\mathbb{E}_{d,d' \sim q(d)}[\mathbb{1}(d = d')] - 1$$

□

[1] Note, under conditional independence, $\beta D_{\mathrm{KL}}(q_\phi(z_{i+1:m} \mid x) \parallel p(z_{i+1:m})) = \sum_{j=i+1}^m \beta D_{\mathrm{KL}}(q_\phi(z_j \mid x) \parallel p(z_j))$

[2] This will be apparent in the following subsection.

Now our goal is to maximize the following objective

$$\begin{aligned}
\mathcal{L}(\theta, \phi) &= I(x;[z,d]) - \beta \mathbb{E}_{x \sim p(x)} D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) \\
&\qquad - \lambda D_{\mathrm{KL}}(q(d) \parallel p(d)) \\
&\geq H(x) + \int \sum_d q(x,z,d) \log q_{\theta,\phi}(x|z,d) dz dx \\
&\quad - \beta \mathbb{E}_{x \sim p(x)} D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) - \lambda D_{\mathrm{KL}}(q(d) \parallel p(d)) \\
&\geq H(x) + \mathbb{E}_{x \sim p(x)}[\mathbb{E}_{z,d \sim q_{\phi,\theta}(\cdot|x)}[\log p_\theta(x|z,d)]] \\
&\quad - \beta \mathbb{E}_{x \sim p(x)} D_{\mathrm{KL}}(q_\phi(z \mid x) \parallel p(z)) \\
&\quad - \lambda(S \mathbb{E}_{d,d' \sim q(d)}[\mathbb{1}(d = d')] - 1)
\end{aligned} \tag{5}$$

4.1. Alternating minimization scheme

Note the lower bound objective in Equation (5) has an inner maximization step over the discrete factors embedded in the variational posterior. Suppose the data is sampled $x^{(i)} \in \mathcal{X}, i = 1, \dots, n$, the continuous latent variables are samples from the decoder $q_\phi(\cdot \mid x^{(i)})$, and the discrete latent variables are represented using one-hot encodings of each variables $d^{(i)} \in \{e_1, \dots, e_S\}$. After rearranging the terms, we arrive at the following optimization problem.

$$\begin{aligned}
\underset{\theta,\phi}{\text{maximize}} & \left(\underbrace{\underset{d^{(1)},\dots d^{(n)}}{\text{maximize}} \sum_{i=1}^n u_\theta^{(i)\intercal} d^{(i)} - \lambda' \sum_{i \neq j} d^{(i)\intercal} d^{(j)}}_{:= \mathcal{L}_{LB}(\theta,\phi)} \right) \\
& \quad - \beta \sum_{i=1}^n D_{KL}(q_\phi(z|x^{(i)})||p(z)) \\
\text{subject to} & \quad \|d^{(i)}\|_1 = 1,\ d^{(i)} \in \{0,1\}^S,\ \forall i,
\end{aligned} \tag{6}$$

where $u_\theta^{(i)}$ denotes the vector of the likelihood $\log p_\theta(x^{(i)}|z^{(i)}, e_k)$ evaluated at each $k \in [S]$. Note, the inner maximization problem $\mathcal{L}_{LB}(\theta, \phi)$ in Equation (6) over the discrete variables $[d^{(1)}, \dots, d^{(n)}]$ subject to the sparsity equality constraints can be exactly solved in polynomial time via *minimum cost flow* without continuous relaxation as shown in Theorem 1 in (Jeong & Song, 2018).

5. Related works

Unsupervised discovery of disentangled factors dates back to 90s. Schmidhuber (1992) penalizes the predictability of a latent dimension given the others but the approach did not scale very well. More recently, VAE was proposed and the framework offered scalability and the optimization stability (Kingma & Welling, 2013). Then Higgins et al. (2017) showed that tuning the β hyperparameter in VAE to $\beta > 1$ can influence the model to learn statistically independent and disentangled representations by limiting the capacity of the latent information channel.

The recent follow up works from Kim & Mnih (2018); Chen et al. (2018); Gao et al. (2018) then analyzed the KL divergence term under the expectation over the data distribution could be broken down into the mutual information term between the data and the latent code, and the KL divergence term between the latent distribution and the factorial prior often denoted as total correlation (TC) (Watanabe, 1960). The idea was that regularizing the KL divergence term with high β in the VAE framework not only penalizes for TC but also inevitably penalizes the mutual information between the data and the latent variables.

To this end, the recent works proposed decreasing β but more explicitly penalizing for TC. FactorVAE estimates the total correlation via density ratio trick which utilizes a discriminator network (Kim & Mnih, 2018). β-TCVAE employs mini-batch weighted sampling to estimate TC (Chen et al., 2018). On the contrary, we observe a factorization $TC(z) = \sum_{i=2}^m I(z_{1:i-1}; z_i)$ and show we can penalize TC without any explicit computation by incrementally penalizing each summand in the factorization by cascading the information flow.

On the other hand, NVIL (Mnih & Gregor, 2014) and VIMCO (Mnih & Rezende, 2016) have explored training VAEs with only discrete latent variables via REINFORCE (Williams, 1992) with variance reduction techniques. VQ-VAE (van den Oord et al., 2017) learns discrete latent variables via vector quantization. However, modeling via binary latent variables alone can be inappropriate since the underlying modalities would be a mix of both continuous and discrete factors for high dimensional complex data.

Jointly modeling the continuous and discrete generative latent factors has been much less explored. InfoGAN (Chen et al., 2016) models both the factors and is based on Generative Adversarial Network (GAN) framework (Goodfellow et al., 2014). InfoGAN aims at disentangling the factors by maximizing the mutual information between a subset of latent dimensions and the generated samples. However, (Kim & Mnih, 2018) showed the learning process can be very unstable and significantly reduces the sample diversity in contrast to the β-VAE framework (Higgins et al., 2017). Empirically, InfoGAN tends to mix discrete and continuous factors which results in lower disentanglement score (Kim & Mnih, 2018) than β-VAE based methods. JointVAE proposed jointly modeling both the continuous and discrete factors by augmenting the continuous and discrete latent variables. This introduces an additional KL divergence term for discrete latent variables which is optimized by a continuous reparameterization via Gumbel softmax trick (Gumbel, 1954; Maddison et al., 2016; Jang et al., 2016). Our method, on the other hand, decouples the task of jointly modeling the continuous and discrete factors of data via alternating maximization and shows significant gains in the disentanglement performance.

6. Implementation details

Model architecture and training details are provided in supplementary B. As discussed in Section 3, we individually control the β term on each continuous variables. Let β_j denote the coefficient for a variable j. Each β_j's start at the high value β_h and gets relieved one at a time to the low value β_l. After each r iterations, we relieve one variable j by switching the coefficient from β_h to β_l. The alternating maximization with discrete variables is enabled after a warm-up time denoted as t_d.

For ablation study, we denote our method without the discrete variables as CascadeVAE-C, our method including the discrete variables but without the information cascading as CascadeVAE-D, and the full method as CascadeVAE. Algorithm 1 shows the pseudocode for CascadeVAE. Note, if t_d is greater than MAXITER, then the pseudocode trains CascadeVAE-C.

Algorithm 1 CascadeVAE

Input : Data $\{x^{(i)}\}_{i=1}^{N}$, Encoder(q_ϕ), Decoder(p_θ), β_l, β_h, r, t_d, optimizer g
Initialize parameters ϕ, θ.
Set $\beta_j = \beta_h, \ \ \forall j$ and $d^{(i)} = 0, \ \ \forall i \in [N]$
Set $j = 1$.
for $t = 1, \ldots,$MAXITER **do**
  if t is a multiple of r
    Switch β_j to β_l and $j \leftarrow j + 1$
  Randomly select batch $\{x^{(i)}\}_{i \in \mathcal{B}}$
  Sample $z_\phi^{(i)} \sim q_\phi(z|x^{(i)}) \ \forall i \in \mathcal{B}$
  if $t > t_d$
    Update $u_\theta^{(i)}$ by computing $\log p_\theta(x^{(i)}|z^{(i)}, e_k) \ \forall k$
    Compute $\mathcal{L}_{LB}(\theta, \phi)$ by solving for the optimal assignment $\{d^{(i)}\}_{i \in \mathcal{B}}$ via minimum cost flow
  $\theta, \phi \leftarrow g\left(\nabla_{\theta,\phi} \mathcal{L}_{LB}(\theta, \phi)\right)$
end for

7. Experiments

We perform experiments on dSprites (Matthey et al., 2017), MNIST, FashionMNIST (Xiao et al., 2017), and Chairs (Aubry et al., 2014) datasets. For quantitative results, dSprites dataset comes labeled with the generative factors which allow quantitative comparisons on the disentanglement metrics. We additionally report unsupervised classification accuracy using the learned discrete variables from running inference on dSprites, MNIST, and FashionMnist.

7.1. Experiments on dSprites

DSprites has $737,280$ images of size 64×64 with 5 generative factors: shape (3), scale (6), orientation (40), x-position (32), and y-position (32). We evaluate the performance with the disentanglement score metric proposed by Kim & Mnih (2018). The details on disentanglement score are provided in Supplementary C. Table 1 compares the disentanglement scores for various baselines. For AnchorVAE, we experimented anchoring 5 latent dimensions out of total 5 and 20 dimensions. For FactorVAE, we performed hyperparameter search over both β and γ to reproduce the reported results from the paper. The dimension of discrete latent representation S, is fixed to 3 following the experiment protocol in JointVAE for a fair comparison.

The results for CascadeVAE-C show that cascading the information flow alone in the β-VAE framework achieves competitive disentanglement scores to the current state of the art FactorVAE method. CascadeVAE-D which does not penalize for TC via information cascading also performs well and boosts the performance of β-VAE by up to 10 points on the disentanglement metric.

Our full method without ablations is denoted as CascadeVAE (the last row) in Table 1. The method shows approximately 10 points boost on top of the current state of the art FactorVAE method. The experiments suggest that both the information cascading for implicit TC penalization and the discrete modeling via alternating disentanglement have complementary benefits leading to a significant improvement over the baseline methods.

Method	m	Mean (std)	Best
β VAE			
($\beta = 10.0$)	5	70.11 (7.54)	84.62
($\beta = 4.0$)	10	74.41 (7.68)	88.38
AnchorVAE			
($\beta_l = 10.0$)	5 (5)	76.36 (4.96)	82.75
($\beta_l = 7.0$)	5 (20)	72.44 (6.85)	83.25
FactorVAE	5	81.09 (2.63)	85.12
	10	82.15 (0.88)	88.25
CascadeVAE-C			
($\beta_l = 0.7$)	5	81.69 (3.14)	88.38
($\beta_l = 1.0$)	10	81.74 (2.97)	87.38
JointVAE	6	74.51 (5.17)	91.75
	4	73.06 (2.18)	75.38
CascadeVAE-D			
($\beta = 7.0$)	6	79.67 (5.36)	90.25
($\beta = 1.0$)	4	80.70 (4.77)	96.50
CascadeVAE			
($\beta_l = 1.0$)	6	90.49 (5.28)	**99.50**
($\beta_l = 2.0$)	4	**91.34 (7.36)**	98.62

Table 1. DSprites disentanglement score for various baselines. The score is obtained from 10 different random seed each with the best hyperparameters.

Figure 2 shows the scatter plot of $TC(z)$ versus the disentanglement scores for 10 different random seeds. We chose the best hyperparameter settings for each method. The results for CascadeVAE-C show comparable $TC(z)$ with β-VAE

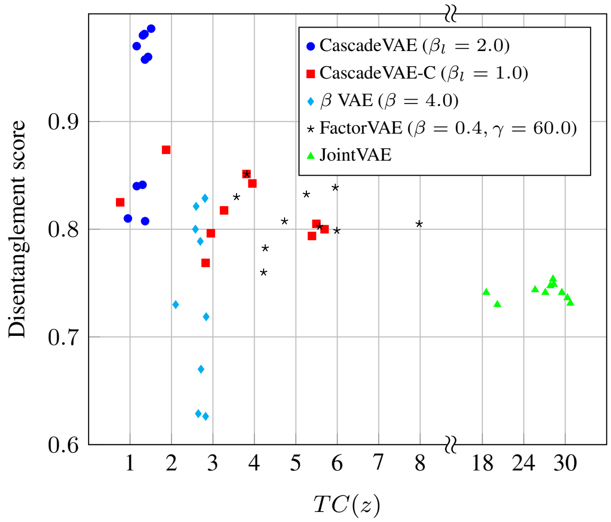


Figure 2. $TC(z)$ and disentanglement score on 10 different random seed on dSprites dataset

even though the β coefficient for our case is much smaller ($\beta_l = 1.0$ for CascadeVAE-C and $\beta = 4.0$ for β-VAE). This also shows we can effectively penalize TC implicitly by information cascade without adding extra discriminator networks for explicit penalization while preserving the disentanglement performance. Notably, CascadeVAE shows smallest TC while showing the highest disentanglement performance, again confirming our hypothesis that both TC and joint modeling of discrete and continuous factors are essential in learning disentangled representations.

In dSprites dataset, the discrete factors encode the categorical shape information (ellipse, heart, square). Following the experiment protocol in JointVAE, we evaluated the classification accuracy computed from the discrete variables from inference. Table 2 compares the results of our method against JointVAE. Note, other baseline methods do not jointly model the continuous and discrete variables. The results show about 30% difference in the classification accuracy. Notably, at the best run, we achieve 99.66% classification accuracy even though the method is unsupervised.

Method	m	Mean (std)	Best
JointVAE	6	44.79 (3.88)	53.14
	4	43.99 (3.94)	54.11
CascadeVAE	6	**78.84 (15.65)**	**99.66**
	4	76.00 (22.16)	98.72

Table 2. Unsupervised classification results on dSprites, $S = 3$. Unsupervised classification accuray for random chance is 33.33.

For qualitative results, Figure 4 first shows the latent traversal results from β-VAE and FactorVAE. The results show that the methods are capable of capturing the x, y-positions, and scale factors but does not disentangle the orientation and shape factors of variation clearly. Figure 3 compares the discrete traversal results against JointVAE. Even though we chose the best runs out of 10 random seeds for both methods, JointVAE does not clearly disentangle the discrete shapes. In contrast, CascadeVAE shows almost perfect disentanglement of the discrete shapes where the discrete code [100], [010], and [001] correspond to the ellipse, heart, and square categories respectively. Figure 5 (left) shows the iteration versus mutual information of each latent dimension $I(x; z_i)$ plot when CascadeVAE is trained with 6 continuous and 1 discrete variables. Figure 5 (right) shows the traversal results for each variables.

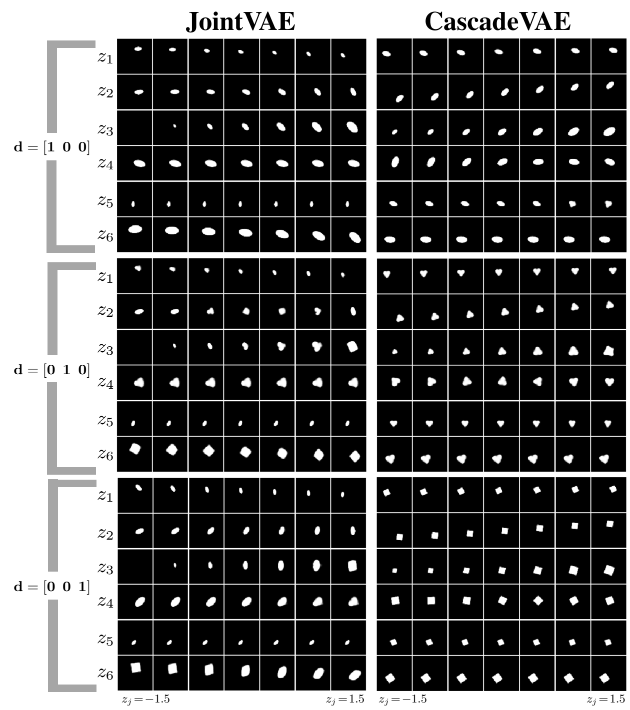


Figure 3. DSprites latent space traversal results. (Left) Best run of Joint VAE with disentanglement score (91.75). (Right) Best run of CascadeVAE with disentanglement score (99.50)

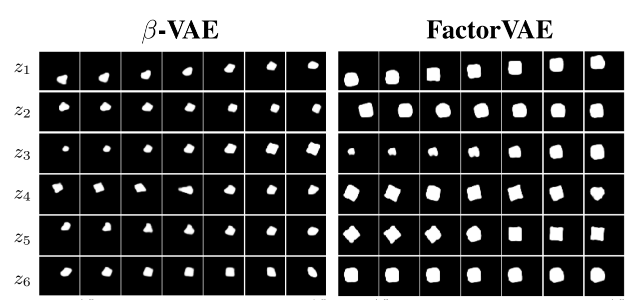


Figure 4. DSprites latent space traversal results. (Left) β-VAE, (Right) FactorVAE

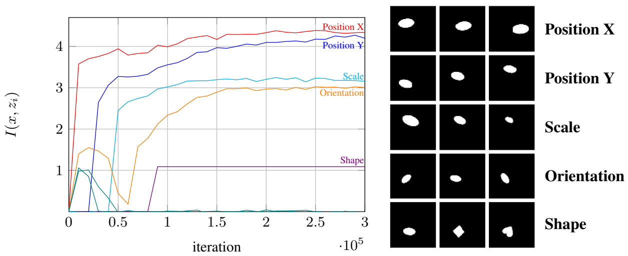


Figure 5. (Left) Increase of mutual information between image and each dimension $I(x, z_j)$ or $I(x, d)$ during training on Dsprites dataset. (Right) Each row respresents latent traversal across each dimension sorted by $I(x, z_j)$. Traversal range is $(-1.2, 1.2)$.

7.2. Experiments on MNIST

MNIST has $60,000$ images of size 28×28. We split $50,000$ as training images and $10,000$ as test images. MNIST dataset has 10 discrete digit categories. For quantitative comparison, Table 3 compares the classification accuracy computed from the discrete variables from inference against JointVAE. The results show that CascadeVAE outperforms JointVAE by 15% classification accuracy. We fixed $S = 10$ following the experiment protocol in JointVAE for a fair comparison.

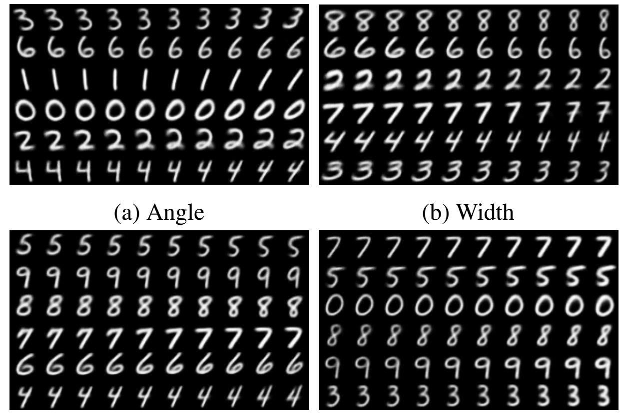
For qualitative results, Figure 6 and Figure 7 show the continuous and discrete latent traversal results, respectively. The results show smooth transitions in angle, width, stroke, and thickness respectively for continuous traversal. The discrete traversal shows that it captures the categorical information of MNIST. Figure 9 (left) shows the iteration versus mutual information of each latent dimension $I(x; z_i)$ plot when CascadeVAE is trained with 10 continuous and 1 discrete variables. Figure 9 (right) shows the traversal results for each variables.

Method	m	Mean (std)	Best
JointVAE	10	68.57 (9.19)	82.30
	4	78.33 (7.18)	92.81
CascadeVAE	10	81.41 (9.54)	**97.31**
	4	**84.19 (5.02)**	96.39

Table 3. Unsupervised classification results on MNIST, $S = 10$. Unsupervised classification accuray for random chance is 10.00.

Method	m (conti)	Mean (std)	Best
JointVAE	10	50.99 (2.5)	55.64
	4	51.51 (4.42)	61.79
CascadeVAE	10	56.01 (3.33)	63.01
	4	**57.72 (3.29)**	**63.55**

Table 4. Unsupervised classification results on FashionMNIST, $S = 10$. Unsupervised classification accuray for random chance is 10.00.



(a) Angle (b) Width (c) Stroke (d) Thickness

Figure 6. Latent traversals on MNIST. Images in a row has the same latent variables except the traversed variable.

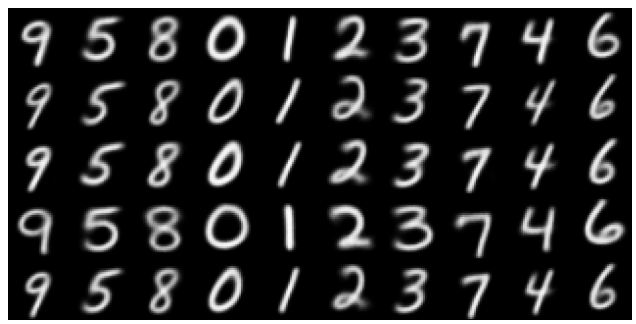
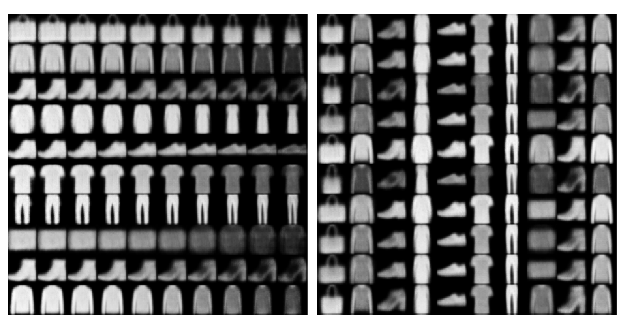


Figure 7. MNIST discrete latent space traversal



(a) Continous (b) Discrete

Figure 8. Latent traversals of the most informative continuous dimension (left) and the discrete dimension (right) on FashionMNIST. Traversal range in continuous dimension is (-2.0, 2.0).

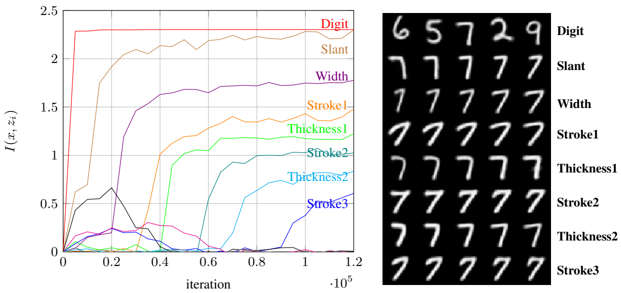


Figure 9. (Left) Increase of mutual information between image and each continuous $I(x, z_j)$ and discrete factors $I(x, d)$ during training on MNIST dataset. (Right) Each row respresents latent traversal across each dimension sorted by $I(x, z_j)$. Traversal range is $(-2.0, 2.0)$.

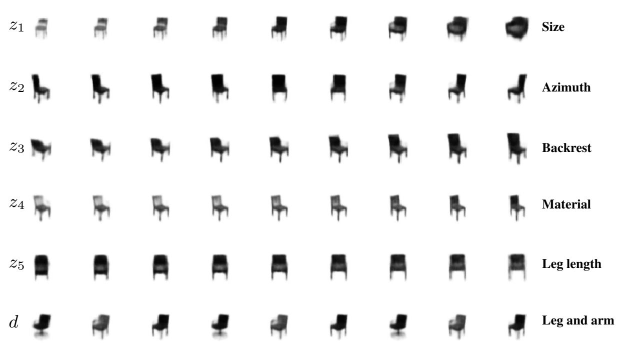


Figure 10. Latent space traversal on chairs dataset. The last row shows the latent traversal of the discrete factor of dimension 3 with the period of 3.

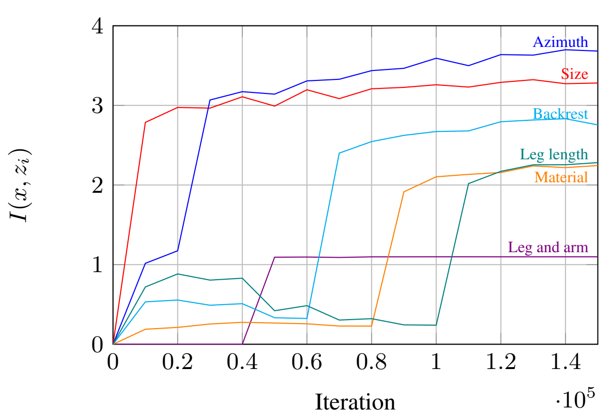


Figure 11. Increase of $I(x, z_i)$ during training the chairs dataset

7.3. Experiments on FashionMNIST

FashionMNIST dataset has $60,000$ images of size 28×28 with 10 categorial labels like MNIST. Table 4 shows the unsupervised classification results from inference in comparison to JointVAE. The results show CascadeVAE consistently outperforms JointVAE. Figure 8 shows the latent traversal of the most informative continuous dimension and discrete dimension.

7.4. Chairs

We preprocessed the chairs dataset (Aubry et al., 2014) and prepared $86,366$ images of size 64×64. Since the chairs dataset is unlabeled in contrast to MNIST, we can only evaluate the qualitative performance instead. Figure 10 shows the latent traversal results on chairs. The traversal on discrete latent variables (the last row in the figure) shows that it captures to the categorial shape of different chairs. Figure 11 shows the iteration versus the estimated mutual information $I(x, z_i)$ plot for each variables.

8. Conclusion

We have developed CascadeVAE for jointly learning the discrete and continuous factors of data in a β-VAE framework. We first propose an efficient procedure for implicitly penalizing the total correlation by controlling the information flow on each variable. This allows us to penalize the total correlation without using extra discriminator networks or sampling procedures. Then, we show a method for jointly learning discrete and continuous latent variables in an alternating maximization framework where we alternate between finding the most likely discrete configurations based on the continuous latent variables, and updating the inference parameters based on the discrete variables.

Our ablation study shows that information cascading and alternating maximization of discrete and continuous variables, provide complementary benefits and leads to the state of the art performance in 1) disentanglement score, and 2) classification accuracy score from the discrete inference network, compared to a number of recently proposed methods.

Acknowledgements

This work was partially supported by Kakao, Kakao Brain and Basic Science Research Program through the National Research Foundation of Korea (NRF) (2017R1E1A1A01077431). Hyun Oh Song is the corresponding author.

References

Aubry, M., Maturana, D., Efros, A. A., Russell, B. C., and Sivic, J. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *TPAMI*, 2013.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in β-vae. In *arXiv preprint arXiv:1804.03599*, 2018.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *NIPS*, 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

Dupont, E. Learning disentangled joint continuous and discrete representations. In *NIPS*, 2018.

Gao, S., Brekelmans, R., Steeg, G. V., and Galstyan, A. Auto-encoding total correlation explanation. *arXiv preprint arXiv:1802.05822*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, 2014.

Gumbel, E. J. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 1954.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jeong, Y. and Song, H. O. Efficient end-to-end learning for quantizable representations. In *ICML*, 2018.

Kim, H. and Mnih, A. Disentangling by factorising. In *ICML*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2013. *URL https://arxiv. org/abs/1312.6114*, 2013.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Mnih, A. and Rezende, D. J. Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*, 2016.

Schmidhuber, J. Learning factorial codes by predictability minimization. *Neural Computation*, 1992.

van den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *NIPS*, 2017.

Watanabe, S. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 1960.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.