Proceedings of Machine Learning Research 129:305–320, 2020 ACML 2020

Disentangled Representations for Sequence Data using Information Bottleneck Principle

Masanori Yamada* MASANORI.YAMADA.CM@HCO.NTT.CO.JP
NTT Secure Platform Laboratories, NTT Corporation, Tokyo, Japan

Heecheol Kim* H-KIM@ISI.IMI.I.U-TOKYO.AC.JP
The University of Tokyo, Tokyo, Japan

Kosuke Miyoshi MIYOSHI@NARR.JP
narrative nights inc., Kanagawa, Japan

Tomoharu Iwata TOMOHARU.IWATA.GY@HCO.NTT.CO.JP
NTT Communication Science Laboratories, Kyoto, Japan

Hiroshi Yamakawa YMKW@WBA-INITIATIVE.ORG
The Whole Brain Architecture Initiative, Tokyo, Japan

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

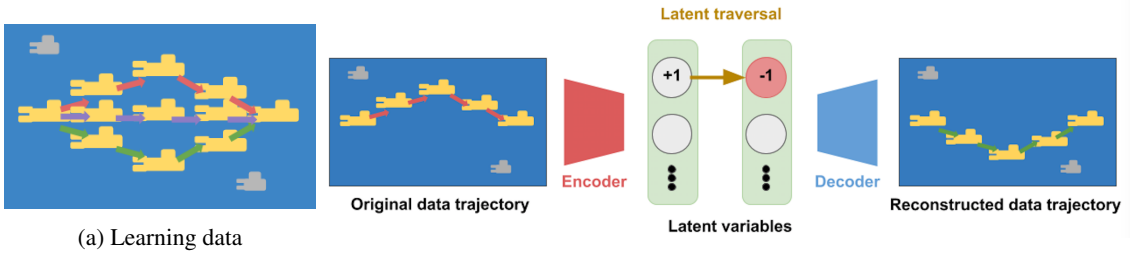
We propose the factorizing variational autoencoder (FAVAE), a generative model for learning disentangled representations from sequential data via the information bottleneck principle without supervision. Real-world data are often generated by a few explanatory factors of variation, and disentangled representation learning obtains these factors from the data. We focus on the disentangled representation of sequential data which can be useful in a wide range of applications, such as video, speech, and stock markets. Factors in sequential data are categorized into dynamic and static ones: dynamic factors are time dependent, and static factors are time independent. Previous models disentangle between static and dynamic factors and between dynamic factors with different time dependencies by explicitly modeling the priors of latent variables. However, these models cannot disentangle representations between dynamic factors with the same time dependency, such as disentangling "picking up" and "throwing" in robotic tasks. On the other hand, FAVAE can disentangle multiple dynamic factors via the information bottleneck principle where it does not require modeling priors. We conducted experiments to show that FAVAE can extract disentangled dynamic factors on synthetic, video, and speech datasets.

Keywords: Representation Learning, Sequential Data

1. Introduction

Representation learning is one of the most fundamental problems in machine learning. A real-world data distribution can be regarded as a low-dimensional manifold in a high-dimensional space (Bengio et al., 2013). Generative models in deep learning, such as the variational autoencoder (VAE) (Kingma and Welling, 2013) and generative adversarial network (GAN) (Goodfellow et al., 2014), can learn a low-dimensional manifold representation as a latent variable. The factors are fundamental components of the data; for example, size, back length, and leg style are the factors of an

* Both authors equally contributed to this paper.



(a) Learning data

(b) Latent traversal of FAVAE

Figure 1: Illustration of latent traversal on a dynamic factor. FAVAE takes into account sequences of data points, taking all data points in a trajectory as one datum. For example, for a pseudo-dataset representing the trajectory of a submarine (1a), FAVAE accepts three different trajectories of the submarine as sequential data. FAVAE learns the factor that controls the trajectory of the submarine, so latent traversal shows changes in the submarine's trajectory (1b).

image in the 3D Chairs dataset (Aubry et al., 2014). Disentangled representation is a state where each factor is acquired as each element in latent variables (Bengio et al., 2013). Thus, if in a model with learned disentangled representation, shifting single latent variable while leaving the others fixed generates data showing that only the corresponding factor is changed. This is called *latent traversals* (a good demonstration of which was given by Higgins et al. (2016a)[1]). Disentangled representation has three advantages. First, latent variables are interpretable. Second, disentangled representation helps improve the generalization performance. Third, disentangled representation is robust against adversarial attacks (Alemi et al., 2016).

We focus on the disentangled representation learning of sequential data. Sequential data are generated by dynamic and static factors: dynamic factors are time dependent, and static factors are time independent. An example of a dynamic factor is the motion of characters in Sprites video dataset, whereas an example of a static factor is hair color (Li and Mandt, 2018). We define static and dynamic factors in Sec. 2. Existing disentangled representation learning models (Higgins et al., 2016a,b; Burgess et al., 2018; Kim and Mnih, 2018; Chen et al., 2018; Kumar et al., 2017; Rubenstein et al., 2018; Chen et al., 2016), for non sequential data cannot extract dynamic factors. The concept of disentangled representation learning for sequential data is illustrated in Fig. 1. Consider that the pseudo-dataset of the movement of a submarine has a dynamic factor: the trajectory shape. The disentangled-representation-learning model for sequential data can extract this shape.

There is a wide range of potential applications if we extend disentanglement representation to sequential data such as speech, video, and stock markets. For example, disentangled representation learning for a video dataset can extract the motion and the change in orientation of a given character in a video dataset. Another application is the reduction of action space in reinforcement learning. Extracting dynamic factors would enable the generation of macro-actions (Durugkar et al., 2016), which are sets of sequential actions that represent the fundamental factors of the actions. Thus, disentangled representation learning for sequential data opens the door to new areas of research.

Recent related work (Hsu et al., 2017; Li and Mandt, 2018) disentangled between static and dynamic factors of sequential data. The factorized hierarchical VAE (FHVAE) (Hsu et al., 2017) is based on a graphical model using latent variables with different time dependencies. By maximizing the variational lower bound of the graphical model, FHVAE can disentangle dynamic factors with

1. This demonstration is available at http://tinyurl.com/jgbyzke

different time dependencies, but it cannot disentangle representations with the same time dependency. The Disentangled Sequential Autoencoder (DSAE) developed by (Li and Mandt, 2018) is the same as that in the FHVAE in terms of the time dependencies of the latent variables. Since these models require different time dependencies for the latent variables, they cannot be used to disentangle different dynamic factors with the same time-dependency. For example, two dynamic factors like goal position and trajectory shape have the same time dependency in the 2D Reaching dataset in Sec. 6.2.

We address this problem by taking a different approach that focuses on information compression. First, we analyze the root cause of disentanglement from the perspective of information theory. The term causing disentanglement is derived from a fundamental rule: reduce the mutual dependence between the input and output of an encoder while keeping the reconstruction of the data. This is called the information bottleneck (IB) principle. Second, we naturally extend this principle from non-sequential data to sequential data. This enables the disentanglement of multiple dynamic factors as a consequence of information compression. In sequential data, a disentangled representation is difficult to learn since not only the feature space but also the time space should be compressed. We use the VAE with a ladder network (Zhao et al., 2017) for disentangling the representation hierarchically for each ladder. Since a time convolution layer abstracts the time correlation, the latent variable on a ladder with a more time convolutional layer learns more abstract representations, such as a direction and a trajectory shape of robot arms in picking tasks. We use a ladder network with time convolution to decompose sequential information hierarchically and make disentanglement easier. We have developed a factorizing variational autoencoder (FAVAE) in which we implemented the concept of IB with ladder networks of time convolution layers to learn a disentangled representation in accordance with the level of data abstraction. Since FAVAE is a more general model and unlike FHVAE and DSAE doesn't have the restriction of a graphical model design to distinguish between different factors with different time dependencies, it can disentangle between dynamic factors even if they have the same time dependency. It can also disentangle between static and dynamic factors.

Our main contributions in this paper are summarized as follows: (i) We formulate static and the dynamic factors. (ii) To tackle disentangled representation of sequence data, we propose the FAVAE based on the information bottleneck framework that adding multiple latent variables through a ladder VAE approach. (iii) We show experimentally that the information bottleneck approach can disentangle between static and dynamic factors and between dynamic factors from sequence data. Also we find the ladder network architecture is effective for disentangling.

2. Static and Dynamic Factors

For the clarity, we define static and dynamic factors. We assume sequential data are generated from static factors $\left\{z_1^{\mathrm{s}}, \cdots, z_{N_{\mathrm{s}}}^{\mathrm{s}}\right\}$ and dynamic factors $\left\{z_1^{\mathrm{d}}, \cdots, z_{N_{\mathrm{d}}}^{\mathrm{d}}\right\}$ as follow

$$
\begin{aligned}
& p\left(x_T, x_{T-1}, \cdots x_0 | z_1^{\mathrm{s}}, \cdots, z_{N_{\mathrm{s}}}^{\mathrm{s}}, z_1^{\mathrm{d}}, \cdots, z_{N_{\mathrm{d}}}^{\mathrm{d}}\right) \\
& = p\left(x_0 | z_1^{\mathrm{s}}, \cdots, z_{N_{\mathrm{s}}}^{\mathrm{s}}, z_1^{\mathrm{d}}, \cdots, z_{N_{\mathrm{d}}}^{\mathrm{d}}\right) \prod_{t=1}^{T} p\left(x_t | x_{t-1}, h_{t-1}, z_1^{\mathrm{s}}, \cdots, z_{N_{\mathrm{s}}}^{\mathrm{s}}, z_1^{\mathrm{d}}, \cdots, z_{N_{\mathrm{d}}}^{\mathrm{d}}\right) \qquad (1)
\end{aligned}
$$

where x_t is the data point at t, hidden state h_t includes information of $x_{t-1}, \cdots, x_0$. The difference between static and dynamic factors is that the static factor can represent as

$$p\left(x_t|x_{t-1}, h_{t-1}, z^{\mathrm{s}}_1, \cdots, z^{\mathrm{s}}_{N_{\mathrm{s}}}, z^{\mathrm{d}}_1, \cdots, z^{\mathrm{d}}_{N_{\mathrm{d}}}\right) = p\left(x_t|x_{t-1}, h_{t-1}, z^{\mathrm{d}}_1, \cdots, z^{\mathrm{d}}_{N_{\mathrm{d}}}\right). \tag{2}$$

The static factor affects only x_0, whereas the dynamic factor affects x_{t-1} to x_t. For example, object color is the static factor and is determined only by the initial frame. On the other hand, an object motion is a dynamic factor and is not determined only by initial frame. To clarify the meaning of the same time dependency. We also define the T' time dependency of factor $z^{\mathrm{d}}_{n_{\mathrm{d}}}$ as

$$p\left(x_T, x_{T-1}, \cdots|z^{\mathrm{d}}\right) = p\left(x_0|z^{\mathrm{d}}_{n_{\mathrm{d}}}\right) \prod_{t=T'+1}^{T} p\left(x_t|x_{t-1}, h_{t-1}\right) \prod_{t=1}^{T'} p\left(x_t|x_{t-1}, h_{t-1}, z^{\mathrm{d}}_{n_{\mathrm{d}}}\right). \tag{3}$$

3. Related Work

3.1. Disentanglement for Non-Sequential Data

The β-VAE (Higgins et al., 2016a,b) is commonly used for learning disentangled representations on the basis of the VAE framework (Kingma and Welling, 2013) for a generative model. The β-VAE is an extension of the coefficient $\beta > 1$ of $D_{\mathrm{KL}}\left(q\left(z|x\right)||p\left(z\right)\right)$ in the VAE, where D_{KL} is the Kullback-Leibler (KL) divergence. The objective function of β-VAE is

$$\mathcal{L}_{\beta-\mathrm{VAE}} = E_{q(z|x)}\left[\log p\left(x|z\right)\right] - \beta D_{\mathrm{KL}}\left(q\left(z|x\right)||p\left(z\right)\right), \tag{4}$$

where x is a data point, z is a latent variable, $\beta > 1$, and $p\left(z\right) = \mathcal{N}\left(0, I\right)$. The first term is a reconstruction term used to reconstruct x, and the second term is a regularization term used to regularize posterior $q\left(z|x\right)$ and $p(z)$ is prior of z. Encoder $q\left(z|x\right)$ and decoder $p\left(x|z\right)$ are modeled with neural networks. The β-VAE promotes disentangled representation learning via $D_{\mathrm{KL}}\left(q\left(z|x\right)||p\left(z\right)\right)$. We discuss it in Sec. 4.1.

3.2. Disentanglement for Sequential Data

Several recently reported models (Hsu et al., 2017; Li and Mandt, 2018) disentangle between dynamic factors with different time in sequential data and between static and dynamic factors such as speech and video (Garofolo et al., 1993; Pearce and Picone, 2002). FHVAE (Hsu et al., 2017) can disentangle dynamic factors with different time dependencies, but it cannot disentangle representations with the same time dependency. The input of FHVAE is a sequential dataset that has been partitioned into segments as $\left\{x^n_{t=1:T_n}\right\}_{n=1}^{N}$, where n is the sequence index, N is the number of sequences, t is the segment index, and T_n is the number of segments in the n-th sequence. Note that the segment refers to a variable shorter than the sequence. The FHVAE is modeled as

$$p\left(\left\{x^n_{t=1:T_n}, z^n_1, z^n_2\right\}, \mu\right) = p\left(\mu\right) \prod_{n=1}^{N} p\left(z^n_1\right) p\left(z^n_2|\mu\right) p\left(x^n_{t=1:T_n}|z^n_1, z^n_2\right), \tag{5}$$

where the z^n_2 has a hierarchical prior $p\left(z^n_2|\mu\right) = \mathcal{N}\left(\mu, \sigma^2_{z_2} I\right)$, $p\left(\mu\right) = \mathcal{N}\left(0, \sigma^2_{\mu} I\right)$. The point is that the only z^n_2 has the prior $p\left(\mu\right)$ that does not depend on a segment. FHVAE disentangles between

factors with different time dependencies by using the different temporal dependent structures of the prior. In speech data, FHVAE encodes sequence latent variable z_2 with a factor of distinguish speakers and sequence segment latent variable z_1 with other residual information (e.g., utterance words).

Similarly, DSAE (Li and Mandt, 2018) imposes different temporal dependent constraints on latent variables via the structure of the graphical model. In DSAE, the decoder is

$$p\left(x_{1:T}, z_{1:T}, z_{\mathrm{s}}\right) = p\left(z_{\mathrm{s}}\right) \prod_{t=1}^{T} p\left(z_t | z_{<t}\right) p\left(x_t | z_t, z_{\mathrm{s}}\right) \tag{6}$$

and the encoder is

$$q\left(z_{1:T}, z_{\mathrm{s}} | x_{1:T}\right) = q\left(z_{\mathrm{s}} | x_{1:T}\right) q\left(z_{1:T} | z_{\mathrm{s}}, x_{1:T}\right), \tag{7}$$

where $z_{<t} = z_1, \cdots, z_{t-1}$ and z_{s} is a time independent latent variable. Both FHVAE and DSAE disentangle between different time dependent factors by changing the time dependency of variables. If dynamic factors have the same time dependency, previous models cannot disentangle dynamic factors. Since FAVAE has no time-dependency constraint of the prior or latent variable and performs disentanglement by using a loss function (see Eq. 12), it can disentangle static and dynamic factors as well as sets of dynamic factors even if they have the same time dependency.

4. Proposed Model: Disentanglement for Sequential Data

4.1. Preliminary: Origin of Disentanglement

To clarify the origin of disentanglement in β-VAE, we explain the regularization term. The regularization term has been decomposed into three terms (Chen et al., 2018; Kim and Mnih, 2018; Hoffman and Johnson, 2016):

$$E_{p(x)}\left[D_{\mathrm{KL}}\left(q\left(z|x\right)||p\left(z\right)\right)\right] = I\left(x; z\right) + D_{\mathrm{KL}}\left(q\left(z\right)||\prod_j q\left(z_j\right)\right) + \sum_j D_{\mathrm{KL}}\left(q\left(z_j\right)||p\left(z_j\right)\right), \tag{8}$$

where I is the mutual information and z_j is the j-th dimension of the latent variable. The second term, which is called "total correlation" in information theory, quantifies the redundancy or dependency among a set of random variables (Watanabe, 1960). The β-TCVAE (Chen et al., 2018) has been experimentally shown to reduce the total correlation causing disentanglement. The third term indirectly causes disentanglement by bringing $q\left(z|x\right)$ close to the independent standard normal distribution $p\left(z\right)$. The first term is mutual information between the data variable and latent variable based on the data distribution. Minimizing the regularization term causes disentanglement but disrupts reconstruction via the first term in Eq. (8). The shift C scheme was proposed (Burgess et al., 2018) to solve this conflict:

$$-E_{q(z|x)}\left[\log p\left(x|z\right)\right] + \beta\left|D_{\mathrm{KL}}\left(q\left(z|x\right)||p\left(z\right)\right) - C\right|, \tag{9}$$

where constant shift C, which is called "information capacity", linearly increases during training. This C can be understood from the IB point of view (Tishby et al., 2000). The VAE can be derived

by maximizing the evidence lower bound (ELBO), but β-VAE can no longer be interpreted as an ELBO once this scheme has been applied. Equation 9 is derived from the IB (Alemi et al., 2016; Achille and Soatto, 2018; Tishby et al., 2000; Chechik et al., 2005). In IB of β-VAE, the essential information is extracted to z by reducing the information of z while keeping the information of x for the reconstruction:

$$\max I(z;x) \qquad \text{s.t. } |I(\hat{x};z) - C| = 0, \tag{10}$$

where x follows the true distribution and $\hat{x}$ follows the empirical distribution created by samples from the true distribution. Solving this equation by using Lagrange multipliers drives the objective function of β-VAE (Eq. (9)) with β as the Lagrange multiplier (details in Appendix B of (Alemi et al., 2016)). In Eq. (9), C prevents $I(\hat{x}, z)$ from becoming zero. The IB is used in the literature of classification tasks; however, the formation can be related to the autoencoding objective (Alemi et al., 2016).

4.2. Loss Function

FAVAE learns disentangled and interpretable representations from sequential data without supervision. We consider sequential data $x_{1:T} \equiv \{x_1, x_2, \cdots, x_T\}$ generated from a latent variable model,

$$p(x_{1:T}) = \int p(x_{1:T}|z)\, p(z)\, dz. \tag{11}$$

For sequential data, we replace x with $(x_{1:T})$ in Eq. 9. The objective function of FAVAE is

$$-E_{q(z|x_{1:T})}\left[\log p(x_{1:T}|z)\right] + \beta \sum_{\tilde{l}} \left| D_{\mathrm{KL}}\left(q(z|x_{1:T}) \,||\, p(z)\right)_{\tilde{l}} - C_{\tilde{l}} \right|, \tag{12}$$

where $\tilde{l}$ is the index of the ladder and $p(z) = \mathcal{N}(0,1)$. The variational recurrent neural network (Chung et al., 2015) and stochastic recurrent neural network (SRNN) (Fraccaro et al., 2016) extend the VAE to a recurrent framework. The priors of both networks are dependent on time. The time-dependent prior experimentally improves the ELBO. In contrast, the prior of FAVAE is time independent like those of the stochastic recurrent network (STORN) (Bayer and Osendorfer, 2014) and Deep Recurrent Attentive Writer (DRAW) neural network architecture (Gregor et al., 2015); this is because FAVAE is for disentangled representation learning rather than density estimation. For better understanding, consider FAVAE from the perspective of IB. As with β-VAE, FAVAE can be understood from the IB principle.

$$\max I(z;x_{1:T}) \qquad \text{s.t. } |I(\hat{x}_{1:T};z) - C| = 0, \tag{13}$$

where $\hat{x}_{1:T}$ follows an empirical distribution. These principles make the representation of z compact, while reconstruction of the sequential data is represented by $x_{1:T}$.

4.3. Ladder Network

FAVAE uses a hierarchical representation scheme inspired by the variational ladder AE (VLAE) (Zhao et al., 2017). Encoder $q(z|x_{1:T})$ within a ladder network is defined as

$$h_l = f_l(h_{l-1}), \tag{14}$$

$$z_l \sim \mathcal{N}(\mu_l(h_l), \sigma_l(h_l)), \tag{15}$$

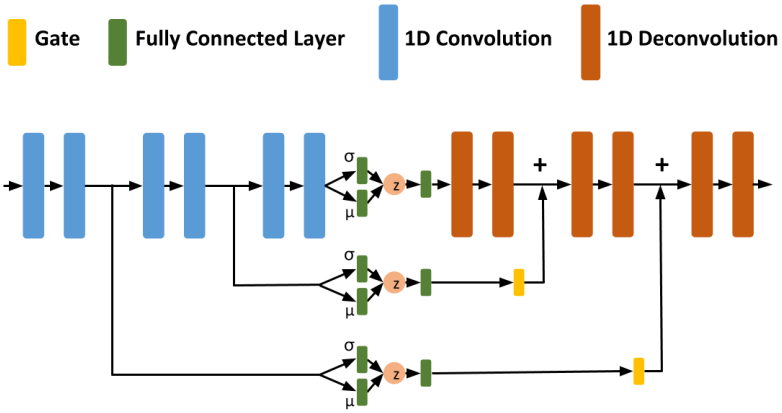


Figure 2: FAVAE architecture

where l is a layer index, $h_0 \equiv x_{1:T}$, and f is a time convolution network, which is explained in the next section. Decoder $p\left(x_{1:T}|z\right)$ within the ladder network is defined as

$$\tilde{z}_L = g_L\left(z_L\right) \tag{16}$$

$$\tilde{z}_l = g_l\left(\tilde{z}_{l+1} + \text{gate}\left(z_l\right)\right), \tag{17}$$

$$x_{1:T} \sim r\left(x_{1:T}, \tilde{z}_0\right), \tag{18}$$

where g_l is the time deconvolution network with $l = 1, \cdots, L-1$, and r is a distribution family parameterized by $g_0\left(\tilde{z}_0\right)$. The gate computes the Hadamard product of its learnable parameter and input tensor. We set r as a fixed-variance factored Gaussian distribution with the mean given by $\mu_{t:T} = g_0\left(\tilde{z}_0\right)$. Figure 2 shows the architecture of FAVAE. The difference between each ladder network in the model is the number of convolution networks through which data pass. The abstract expressions should differ between ladders since the time convolution layer abstracts sequential data. Without the ladder network, FAVAE can disentangle only the representations at the same level of abstraction; with the ladder network, it can disentangle representations at different time correlation of abstraction.

4.4. Time Convolution

There are several mainstream neural network models designed for sequential data, such as the long short-term memory (LSTM) model (Hochreiter and Schmidhuber, 1997), gated recurrent unit model (GRU) (Chung et al., 2014), and quasi-recurrent neural network (QRNN) (Bradbury et al., 2016). Sequential data need a large temporal length for representing the complex motion of multiple dynamic factors. Since time convolution is faster and more stable for long sequential data than LSTM and GRU, we use a time convolutional network. In fact, LSTM and GRU cannot learn stably with the datasets (2D Reaching, 2D Wavy Reaching) used in our experiments. The input data are $x_{t,i}$, where t is the time index and i is the feature index. The time convolution takes into account the dimensions of feature vector j as a convolution channel and performs convolution in the time direc-

tion: $z_{tj} = \sum_{p,i} x_{t-p,i} h_{pij} + b_j$, where j is the channel index. FAVAE has a network similar to that of VLAE regarding time convolution and a loss function similar to that of β-VAE (Eq. (12)).

5. Measuring Disentanglement

Although latent traversals are useful for qualitatively checking the success or failure of disentanglement, the disentanglement needs to be quantified for reliably evaluating a learned model. Various disentanglement quantification methods have been proposed (Do and Tran, 2019; Eastwood and Williams, 2018; Chen et al., 2018; Kim and Mnih, 2018; Higgins et al., 2016b,a; Ridgeway and Mozer, 2018; Kumar et al., 2017), but there is no standard method. All methods basically measure the degree of disentanglement from the relationship between the latent variable and factor. A large-scale experimental study reports that the same trend is observed regardless of these disentanglement metrics (except for Modularity (Ridgeway and Mozer, 2018)) (Locatello et al., 2018). In this paper, we use the mutual information gap (MIG) (Chen et al., 2018) as the metric for disentanglement. The basic idea of MIG is measuring the mutual information between latent variables z_j and a ground-truth factor z_k^*. Higher mutual information means that z_j contains more information regarding z_k^*.

$$\mathrm{MIG} \equiv \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H\left(z_k^*\right)} \left(I\left(z_{j^{(k)}}; z_k^*\right) - \max_{j \neq j^{(k)}} I\left(z_j; z_k^*\right) \right), \tag{19}$$

where K is the number of factors, $j^{(k)} \equiv \arg\max_j I\left(z_j; z_k^*\right)$, and $H\left(z_k^*\right)$ is entropy for normalization.

MIG has a problem when measuring with simple data. When reconstruction is possible with one latent variable, using large β gathers all factor information in one latent variable and MIG becomes large. For example, when goal position, curved inward/outward, and degree of curvature cannot be disentangled to different latent variables in the 2D Reaching dataset, MIG can become large. In our experiments we avoided this problem by excluding the case in which all factor information concentrates in one latent variable.

6. Experiments

We experimentally evaluated FAVAE using six sequential datasets: two bi-dimensional space reaching (2D Reaching, 2D Wavy Reaching) datasets with sequence lengths of 100 and 1000, video dataset Sprites (Li and Mandt, 2018), and speech dataset Timit (Garofolo et al., 1993). The 2D Reaching and 2D Wavy Reaching datasets contain multiple dynamic factors, and the Sprites and Timit datasets contain multiple static factors and a dynamic factor. Table 1 shows all factors in all datasets.

We used FAVAE, FHVAE, and DSAE in all datasets. The FAVAE has several options, i.e., FAVAE with $\beta = 0$, FAVAE with/without the ladder network (L) and information capacity (C). We represent these options as (L, C). (L) means with ladder and (C) means with information capacity. For example, FAVAE (L-) means FAVAE with a ladder network without information capacity.

6.1. Setup

For the training, we used a batch size of 128 and the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3}. FAVAE consists of three ladder networks, which have two time convolu-

Table 1: Table of ground-truth factors in 2D Reaching, 2D Wavy Reaching, Sprites, and Timit. Corresponds to the 2D Reaching, 2D Wavy Reaching, Sprites, and Timit dataset from the top row to bottom row.

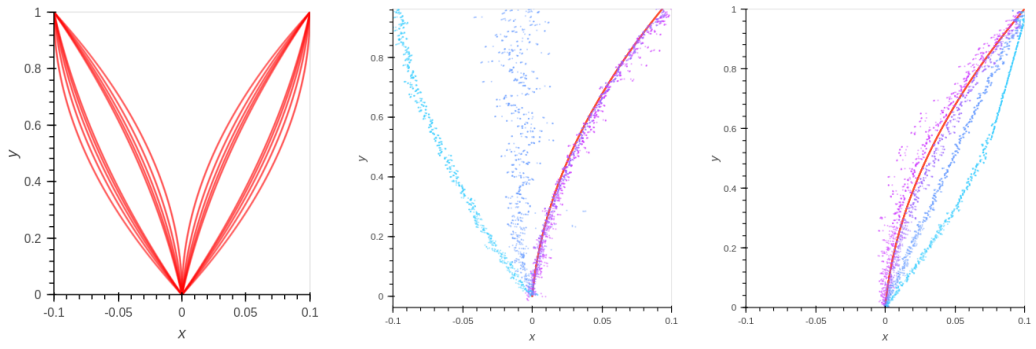
	Static/Dynamic (degree of freedom) :Ground-truth factor
2D Reaching	
Factor 1	Dynamic (2): Goal position
Factor 2	Dynamic (2): Curved inward / outward
Factor 3	Dynamic (5): Degree of curvature
2D Wavy Reaching	
Factor 1	Dynamic (5): Goal position (up left or up right)
Factor 2	Dynamic (4): 1st trajectory shape
Factor 3	Dynamic (4): 2nd trajectory shape
Factor 4	Dynamic (2): 1st trajectory degree of curvature
Factor 5	Dynamic (2): 2nd trajectory degree of curvature
Sprites	
Factor 1	Static (2): Skin color (light or dark2)
Factor 2	Static (2): Shirts color (brown or teal)
Factor 3	Static (2): Hair color (green or pink)
Factor 4	Static (2): Pants color (red or teal)
Factor 5	Static (3): Character Direction (left, forward, right)
Factor 6	Dynamic(3): Motion (Spellcast, Walk, Slash)
Timit	
Factor 1	Static (8): location
Factor 2	Static (2): sex
Factor 3	Static (630): people
Factor 4	Dynamic (2): sentence

tion layers and two time deconvolution layers as shown in Fig. 2. We used Batch Normalization, LeakyReLU with negative slope 10^{-2}, and tanh for the output layer. In the case without a ladder network, we use six time convolution layers and time deconvolution layers the same as the case with a ladder network for a fair comparison. The dimensions of latent variables are $8, 4, 2$ in the lower, middle, and higher ladders. In the case without a ladder network, the dimension of latent variable is 14 for a fair comparison with the case with a ladder network.

Since C is the capacity of information necessary for reconstruction, we used the value of KL divergence loss when learning to reconstruct the data at $C = 0$ as C. We used the best β for maximizing MIG in all experiments. Search space for β and best β are listed in supplementary materials. The encoder used a Gaussian distribution, and the decoder used mean square loss. Since the original implementation of FHVAE used a Gaussian distribution loss in the decoder, we searched for the Gaussian distribution loss and the mean square loss to maximize MIG in FHVAE. To fairly compare reconstruction loss, we evaluated mean square loss even when the Gaussian distribution was used for the decoder. The calculation of MIG uses $10,000$ samples. The setup of FHVAE and DSAE is the same as in the original paper.

6.2. Latent Traversal in 2D Reaching

To demonstrate the disentanglement of the dynamic factors, we used the 2D Reaching dataset. As shown in Fig. 3a, starting from point $(0, 0)$, the point travels to goal position $(-0.1, +1)$ or



(a) (b) (c)

Figure 3: Visualization of latent traversal of FAVAE. 3a represents all data trajectories of 2D Reaching. 3b and 3c represent latent traversal in FAVAE. Each latent variable is traversed, and purple and/or blue points are generated. The color corresponds to the value of the traversed latent variable.

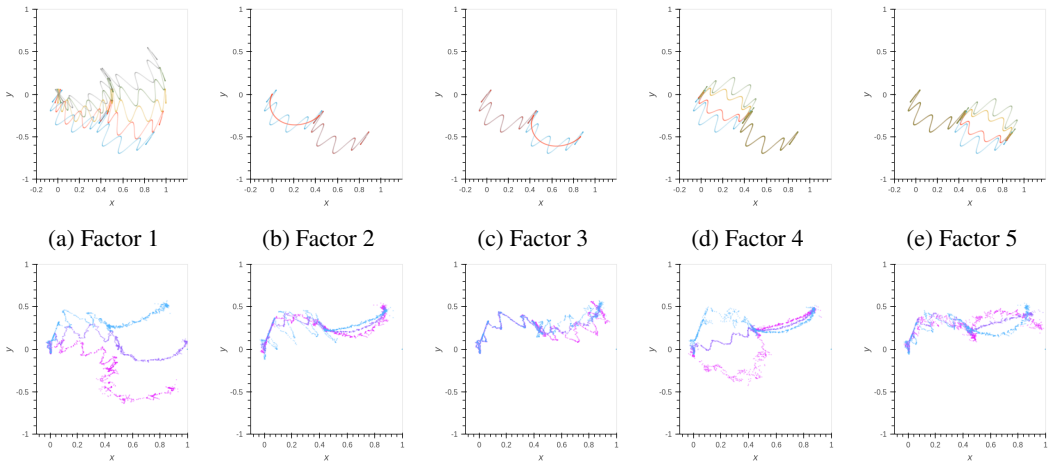
Table 2: Disentanglement scores (MIG) and reconstruction loss (Rec) with standard deviations by repeating 10 times on the different initializations. The standard deviation is in parentheses (e.g., $0.12(9)$ means 0.12 ± 0.09, $0.006(4)$ means 0.006 ± 0.004 and $11.881(24014)$ means 11.881 ± 24.014). The standard deviation is in parentheses. Best results are shown in bold. (L) means with ladder, and (C) means with information capacity (e.g., FAVAE (L-) means FAVAE with a ladder network without information capacity).

Model	2D Reaching				2D Wavy Reaching			
	length=100		length=1000		length=100		length=1000	
	MIG	Rec	MIG	Rec	MIG	Rec	MIG	Rec
FHVAE	0.12(9)	0.01(2)	-	-	0.06(2)	0.171(29)	-	-
DSAE	0.20(16)	0.54(13)	-	-	0.10(1)	2.52(4)	-	-
FAVAE (L) ($\beta = 0$)	0.06(3)	0.022(22)	0.05(4)	0.493(790)	0.02(1)	**0.015(5)**	0.04(3)	**0.085(17)**
FAVAE (- -)	0.07(12)	0.257(173)	0.46(18)	2.209(1869)	0.66(15)	0.041(8)	**0.47(18)**	11.881(24014)
FAVAE (- C)	0.09(13)	0.257(172)	0.46(18)	1.193(1274)	**0.67(16)**	0.042(21)	0.31(10)	5.937(18033)
FAVAE (L -)	**0.28(21)**	**0.006(4)**	0.43(6)	0.022(9)	0.29(9)	0.123(16)	0.28(4)	0.707(86)
FAVAE (L C)	**0.28(11)**	0.008(14)	**0.64(6)**	**0.017(6)**	0.42(17)	0.046(11)	0.24(7)	0.190(95)

$(+0.1, +1)$. There are ten possible trajectories to each goal: five are curved inward, and the other five are curved outward. The degree of curvature for all five trajectories is different. The number of factor combinations was thus 20 ($2 \times 2 \times 5$). The trajectory lengths were 100 and 1000. Figures (3b, 3c) show the results of latent traversal with FAVAE on the 2D Reaching dataset, which were transforming single dimension of latent variable z into another value and reconstructing the output data from the traversed latent variables. FAVAE learns through one entire trajectory and can encode disentangled representations effectively so that feasible trajectories are generated from traversed latent variables (3b, 3c).

6.3. Latent Traversal in 2D Wavy Reaching

To confirm the effect of disentanglement through the IB, we evaluated the validity of FAVAE under more complex factors by adding more factors than 2D Reaching. Five factors in total generated data compared with the three factors that generate data in 2D Reaching. This modified dataset differed in that four of the five factors affect only part of the trajectory: two affected the first half,



(a) Factor 1 (b) Factor 2 (c) Factor 3 (d) Factor 4 (e) Factor 5

(f) Factor 1 (g) Factor 2 (h) Factor 3 (i) Factor 4 (j) Factor 5

Figure 4: Visualization of training data (4a to 4e) and latent traversal (4f to 4j) for 2D Wavy Reaching. The vertical and horizontal axes represent coordinates. Factors $1, 2, 3, 4$, and 5 respectively correspond to Goal position, 1st trajectory shape, 2nd trajectory shape, 1st trajectory degree of curvature, and 2nd trajectory degree of curvature. Each plot was decoded by traversing one latent variable; different colors represent trajectories generated from different values of the same latent variable, z.

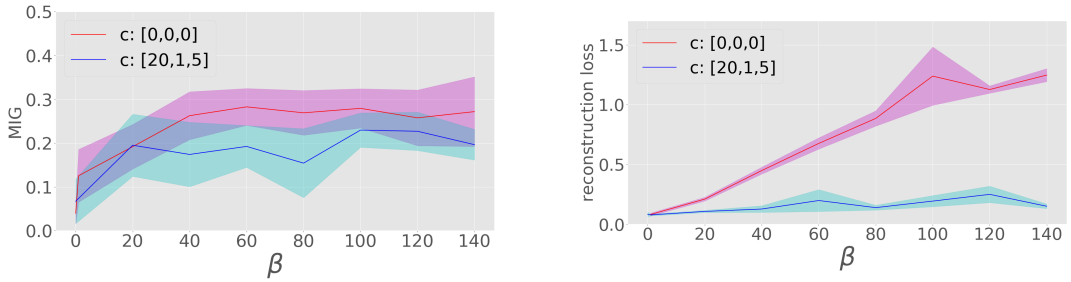


Figure 5: MIG scores and reconstruction losses for different β in FAVAE with ladder network and C. Blue line represents results with information capacity C greater than zero; red line represents results with C set to zero. Note that x axis is plotted in log scale.

and the other two affected the second half. This means that the model should be able to focus on a certain part of the whole trajectory and extract factors related to that part. These factors is explained in detail in Table 1. We show the training dataset of 2D Wavy Reaching and latent traversal in FAVAE with sequence length 1000 in Fig. 4. The latent traversal results for 2D Wavy Reaching are plotted in Figs. 4f to 4j. Even though not all learned representations were perfectly disentangled, the visualization shows that all five generation factors were learned from five latent variables; the other latent variables did not learn any meaningful factors, indicating that the factors could be expressed as a combination of five "active" latent variables.

Table 3: For each factor, counting the number of latent variables z_j that is the highest $I(z_k^*, z_j)$ in each ladder (Lower, Middle, Upper). The same operation is performed ten times in 2D Wavy Reaching, and results are shown. The details of factors are shown in Table 1

	Lower	Middle	Upper
Factor 1	3	0	**7**
Factor 2	**8**	0	2
Factor 3	**8**	0	2
Factor 4	**9**	1	0
Factor 5	**9**	0	1

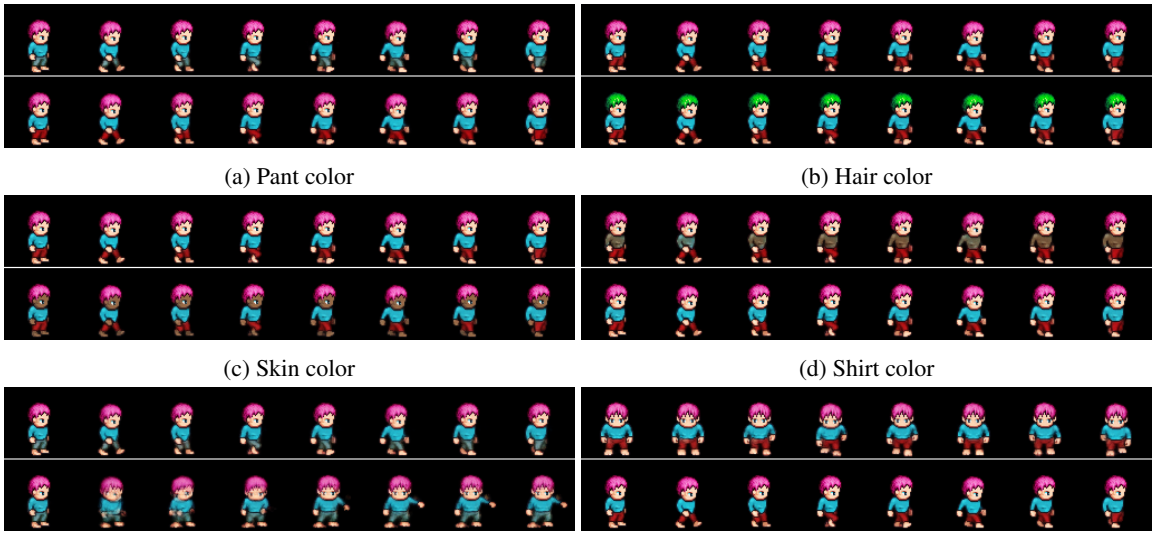
6.4. Disentanglement Between Dynamic Factors

To evaluate the disentanglement performance between dynamic factors, Table 2 compares MIG scores and reconstruction losses for 2D Reaching and 2D Wavy Reaching each with sequence lengths of 100 and 1000. We compared various models on the basis of MIG to demonstrate the validity of FAVAE, i.e., FAVAE with $\beta = 0$, FAVAE with/without the ladder network (L) and information capacity (C), FHVAE (Hsu et al., 2017), and DSAE (Li and Mandt, 2018). FHVAE and DSAE, which are recently proposed disentangled representation learning models, are used as the baseline. Note that FHVAE uses sequence index information (different from the dynamic factor index) to disentangle time series data, which is a different setup from FAVAE and DSAE.

In 2D Reaching and 2D Wavy Reaching, the MIG of the FAVAE was the largest. Even when there were multiple dynamic factors such as in 2D Reaching and 2D Wavy Reaching, FAVAE exhibited good disentanglement performance (large MIG and small reconstruction loss). When the ladder was added, the reconstruction loss was stable (especially at sequence length 1000). For example, looking at the length = 1000 of 2D Wavy Reaching in Table 2, FAVAE without ladder had a large MIG, but the distribution of reconstruction loss was very large.

To confirm the effect of C, we the evaluated reconstruction losses and MIG scores for various β using three ladder networks (Fig. 2) with a different C for each ladder in Fig. 5. We represent each ladder of C as $C = [\mathrm{LowerLadder}, \mathrm{MiddleLadder}, \mathrm{HigherLadder}]$. One setting was $C = [0, 0, 0]$, meaning that C was not used; another setting was $C = [20, 1, 5]$, meaning that C was adjusted on the basis of KL-Divergence for $\beta = 1$ and $C = [0, 0, 0]$. When C was not used, FAVAE could not reconstruct data when β was high; thus, disentangled representation was not learned well when β was high. When C was used, the MIG score increased with β while reconstruction loss was suppressed.

We expect the ladder network can disentangle representations at different levels of abstraction. We evaluated the factor extracted in each ladder by using 2D Wavy Reaching which are generated by different time dependent factors. Factor 1 affects the entire trajectory and four factors (Factor 2,3,4 and 5) affect only half part of the trajectory. Table 3 shows the number of counted values of the index of the latent variable with the highest mutual information in each ladder network. In Table 3, the rows represent factors, and columns represent the index of the ladder networks. Factor 1 (goal position) were extracted the most frequently in the latent variable in the upper ladder. Since the latent variables have eight dimensions for the lower ladder, four dimensions for the middle ladder, and two dimensions for the upper ladder, the upper ladder should be the least frequent when factors are randomly entered for each z. In 2D Wavy Reaching, the goal of the trajectory that affects the



(a) Pant color (b) Hair color

(c) Skin color (d) Shirt color

(e) Motion (f) Direction of character

Figure 6: Visualization of latent traversal of FAVAE on Sprites. Horizontal axis represents sequence, and vertical axis represents differences in z.

Table 4: Disentanglement scores (MIG) and reconstruction loss (Rec) with standard deviations (except FHVAE in Timit) by repeating 10 times on the different initializations. The standard deviation is in parentheses (e.g., $0.15(4)$ means 0.15 ± 0.04). Best results are shown in bold.

Model	Sprites		TIMIT	
	MIG	Rec	MIG	Rec
FHVAE	0.00027(8)	6671.7(81)	0.01(-)	414(-)
DSAE	0.15(4)	1845(1880)	0.01(1)	218(28)
FAVAE (- -)	0.09(4)	644(171)	**0.09(1)**	190(8)
FAVAE (- C)	0.27(15)	1012(582)	0.05(1)	**67(2)**
FAVEA (L -)	0.24(12)	283(37)	0.05(2)	147(13)
FAVAE (L C)	**0.30(8)**	**255(11)**	**0.09(2)**	77(6)

entire trajectory tended to be expressed in the upper ladder. Only Factor 1 represents goal positions, whereas the others represent the shape of a part of the trajectories. Since Factor 1 has a different abstraction level from others, it and the others result in different ladders, e.g., lower ladder and others.

6.5. Disentanglement Between Static and Dynamic Factors

To evaluate the disentanglement performance between static and dynamic factors, we used Sprites and Timit datasets, which contain one dynamic factor and multiple static factors.

Sprites: The Sprites dataset is a video dataset, which was used by Li and Mandt (2018). This dataset contains $3 \times 64 \times 64$ RGB video data with sequential length $= 8$ and consists of five static factors and one dynamic factor (see Table 1). Note that motions are not created with the combination

of dynamic factors, and each motion exists individually (dataset details are explained in Github[2]). We added a Gaussian noise with mean 0 and variance 0.5 to the rescaled input image from -1 to 1. We executed disentangled representation learning by using FAVAE with $\beta = 20$, $C = [20, 10, 5]$.

Timit: To evaluate the effectiveness on a speech dataset, we trained FAVAE with the Timit that is transformed by pre-processing: the raw speech waveforms are first split into sub-sequences of 200ms, and then preprocessed with sparse fast Fourier transform to obtain a 200 dimensional log-magnitude spectrum, computed every 10ms (Garofolo et al., 1993). Timit has four factors (location, sex, people, and sentence), sequential length 536 and feature dimension 80.

Results: Figure 6 shows the results of latent traversal in the Sprite, and we use two z values between -3 and 3. The latent variables in the lower ladder extract expressions of motion (4th z in lower ladder), pant color (5th z in lower ladder), direction of character (6th z in lower ladder) and shirt color (7th z in lower ladder). The latent variables in the middle ladder extract expressions of hair color (1st z in middle ladder) and skin color (2nd z in middle ladder). FAVAE can extract the disentangled representations between static and dynamic factors in high dimension datasets. Table 4 shows MIG and reconstruction loss at Sprites and Timit . FHVAE was unstable in Timit and could not estimate an error because 10 statistics could not be accumulated even when experimenting with 25 different initializations. FAVAE has the largest MIG in both Sprite and Timit and has smaller reconstruction loss than the existing methods. Since some factors (location and people) were difficult to extract, all models had small MIG in Timit.

7. Summary and Future Work

We proposed a factorizing variational autoencoder (FAVAE) for learning disentangled representations via the information bottleneck from sequential data. The experiments using six sequential datasets demonstrated that our FAVAE can learn disentangled representations between different dynamic factors and between static and dynamic factors. In future work, we will apply the FAVAE to actions of reinforcement learning to reduce the pattern of actions.

References

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

2. https://github.com/jrconway3/Universal-LPC-spritesheet

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*, 2016.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for Gaussian variables. *Journal of Machine Learning Research*, 6(Jan):165–188, 2005.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2980–2988, 2015.

Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations. *arXiv preprint arXiv:1908.09961*, 2019.

Ishan P Durugkar, Clemens Rosenbaum, Stefan Dernbach, and Sridhar Mahadevan. Deep reinforcement learning with macro-actions. *arXiv preprint arXiv:1606.04615*, 2016.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. 2018.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pages 2199–2207, 2016.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. 1993.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*, 2016a.

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. 2016b.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Matthew D Hoffman and Matthew J Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems*, pages 1878–1889, 2017.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

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

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

Yingzhen Li and Stephan Mandt. A deep generative model for disentangled representations of sequential data. *arXiv preprint arXiv:1803.02991*, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

David Pearce and J Picone. Aurora working group: DSR front end LVCSR evaluation AU/384/02. *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, pages 185–194, 2018.

Paul K Rubenstein, Bernhard Schölkopf, and Ilya Tolstikhin. Learning disentangled representations with wasserstein auto-encoders. 2018.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017.