Quantitative Comparison of Constrained Variational Autoencoders Regarding their Ability to Learn Disentangled Latent Space Features

Jakob Hartmann
Technische Universität Berlin
jakob.hartmann@campus.tu-berlin.de

Abstract

The research area of learning disentangled representations has become increasingly important in recent years. It has been found that models with a disentangled latent space are better suited for transfer learning and that their output can be interpreted more easily by humans. This seminar paper will first give a general introduction to deep generative models, latent variables, and disentanglement, before different disentanglement metrics and constrained Variational Autoencoders are explained in detail. Afterwards the disentanglement metrics will be used to quantitatively compare the constrained Variational Autoencoders regarding their ability to learn disentangled latent space features. It is found that, of the approaches considered, an objective function which penalizes the Total Correlation between the latent space variables is best suited for learning disentangled representations, but that other factors such as hyperparameters have a large influence on the result.

1 Deep Generative Models

Deep Generative Models (DGMs) are a class of generative models that use neural networks to learn to model the underlying probability distribution of the given training data [1]. As Soleinmany explains in her lecture on Deep Generative Modeling [2], DGMs are part of the unsupervised learning approaches because their input only consists of the training data and no corresponding labels are given. DGMs can learn the probability distribution either explicitly or implicitly: explicit models try to learn the parameters of the probability distribution, which can then be used to generate new samples similar to the training data. Implicit models, on the other hand, attempt to generate new samples directly, without first learning the parameters of the underlying probability distribution. However, generating new samples is not the only goal of DGMs. Like many other unsupervised learning approaches, they are also used to learn more about the underlying structure of the data.

The main area of application of DGMs today, and the focus of this seminar paper is image generation. Here, the models get a large number of images as input and the goal is to generate new samples that are indistinguishable from real ones or to learn more about the underlying features of the images. The most prominent approaches in Deep Generative Modelling are Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) [3]. GANs are typically the choice for generating new images due to their impressive sample quality, while VAEs are at the core of representation learning because of their explicit representation of the latent space. This seminar paper will deal exclusively with VAEs.

2 Latent Variables

A latent variable is a variable that cannot be observed directly but is the underlying factor for other observable variables, which can be used to infer information about the latent variable [2]. An example of a latent variable could be the qualification of an applicant for a position in a company. Since the qualification cannot be determined directly, the company uses other information such as university degrees, reference letters from previous employers or the performance in an assessment center to infer it.

All latent variables combined form the latent space. In his article "Understanding Latent Space in Machine Learning" Tiu [4] explains that this latent space can be understood as a low dimensional representation of the original high dimensional data. DGMs try to learn this underlying latent space representation from the given training data. A DGM for example that learns to generate images of human faces could learn a latent space representation that includes factors such as gender, hair color and presence of glasses. Faces that look similar, meaning that they are close to each other in the original high dimensional space, should also be close to each other in the latent space. Learning a good latent space representation allows, amongst other things, to compress the data and to interpolate within the latent space. Given the corresponding latent space representation of a woman with fair hair and a woman with black hair, a woman with brown hair could be generated.

3 Disentanglement

However, the goal of many DGMs is not only to learn this latent space representation, but also to ensure that it satisfies an important property called disentanglement. In a disentangled latent space representation, each learned latent variable corresponds to exactly one true underlying factor. A true underlying factor is one that explains the variance in the data [5]. For a synthetic dataset like dSprites [6], this could be the shape or size of an object, while for pictures of people it might be the gender. Thus, in a disentangled representation "a change in a single underlying factor of variation z_i, should lead to a change in a single factor in the learned representation $r(\mathbf{x})$" [7, p. 1]. Using a DGM, which was trained to generate images of human faces, this would allow individual features (e.g., hair length) to be changed while all other factors (e.g., hair color) remain the same.

Kumar et al., 2018 [8] highlight several other advantages that reach far beyond image editing. Models which learn disentangled representations can for example be useful for transfer learning. Other models that build on the disentangled representation are faster to train and, maybe even more importantly, require less training data. Moreover, a disentangled latent space allows humans to better interpret the outputs of a model and understand on which basis decisions are made. This could be valuable in medicine or security relevant areas where humans want to verify the decisions of a machine learning model. In recent years, several DGMs, in particular VAEs, have been introduced, which were especially developed to favor the learning of disentangled latent space representations [8]–[12].

The explanation of disentanglement given above is intuitive and used in many papers, but also has its drawbacks. Kim et al., 2018 note, that this "does not allow correlations among the factors or hierarchies over them. Thus this definition seems more suited for synthetic data with independent factors of variation than to most realistic data sets." [11, p. 4]

3.1 Measuring disentanglement

An important question that arises and one which will be relevant when comparing different models is about measuring disentanglement: How well is the learned latent space representation of a model disentangled? To answer this, either a qualitative or quantitative approach can be taken.

The most prominent qualitative approach is called latent traversal and is explained by Morton in his article "Learning Disentangled Representations" [13]. Starting from a latent space representation, one dimension/latent variable is changed several times and given to a DGM (e.g., the decoder of a VAE) and the obtained outputs are evaluated by a human. If the output also changes only along a single dimension (e.g., only the y-position of the white dot in Figure 1 is affected by the change in the latent space), then the model has learned a well disentangled representation. This approach can be applied to all datasets but is very time-consuming and highly subjective and therefore not suitable for the comparison of several models [11].

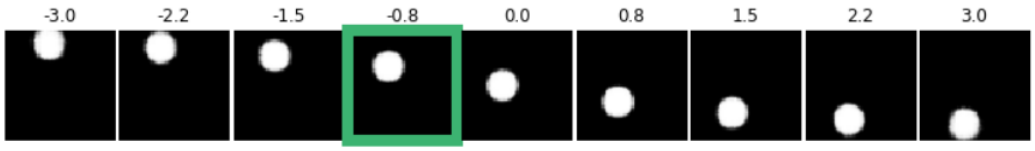

Figure 1: Example of a latent traversal on a simple dataset [13]

To avoid these disadvantages, a quantitative approach can be taken, which is typically based on so-called disentanglement metrics. These metrics measure the difference between the learned latent space features of a model and the true underlying latent factors. They enable an objective comparison of different models, for which no human supervisor is needed. However, there is a major drawback to them: the true underlying latent factors, i.e., the generative factors of the data must be known. This severely limits the number of possible datasets that can be used.

Currently, there is not one commonly used disentanglement metric, but many different ones, most of which have been introduced together with a new form of VAE [8], [9], [11], [12], [14]. Quantitative measurement of disentanglement, especially on datasets where the true underlying latent factors are not known, therefore remains an active area of research in representation learning [11]. In the following, five popular disentanglement metrics, all of which will also be used for the comparison at the end of this paper, are explained in detail.

3.2 Disentanglement metrics

3.2.1 BetaVAE metric

Higgins et al., 2017 [9] were among the first to introduce a metric to quantitatively measure the ability of a model to learn disentangled representations. They argue that a learned latent representation should be independent and interpretable and propose a linear classifier to measure these properties. First, one true underlying latent factor (e.g., scale) is selected and random samples are generated with this factor being fixed. These samples are then fed to a VAE encoder to obtain the corresponding latent representations. Afterwards the pairwise absolute difference between two latent vectors is calculated and then the mean over many of these differences is taken. This mean serves as an input to a linear classifier, which should predict the fixed latent factor. If the VAE learned a well disentangled latent representation, the variance of the fixed variable will be smaller than the variance of the other latent variables, thus making the prediction for the linear classifier easier. The accuracy of this classifier is the disentanglement score for the model. In this seminar paper the name "BetaVAE metric/score" from Locatello et al., 2019 [7] will be used to refer to this metric.

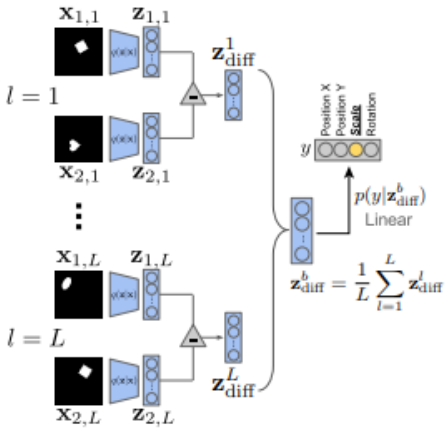

Figure 2: Calculation of the BetaVAE metric [9, Fig. 5]

3.2.2 FactorVAE metric

Kim et al., 2019 [11] identified multiple weaknesses of the BetaVAE metric, especially that it can assign a "100% accuracy even when only K – 1 factors out of K have been disentangled" [11, p. 4]. To calculate their improved disentanglement metric, they also fix a true underlying latent factor, generate random samples, and feed them into a VAE encoder. The resulting latent space representations are then normalized using the empirical standard deviation and afterwards the empirical variance of each latent space dimension across the different representations is calculated. The dimension with the smallest variance then serves as input to a majority vote classifier. The disentanglement score results from the accuracy of this classifier predicting the fixed latent variable. In this seminar paper the name "FactorVAE metric/score" from Locatello et al., 2019 [7] will be used to refer to this metric.

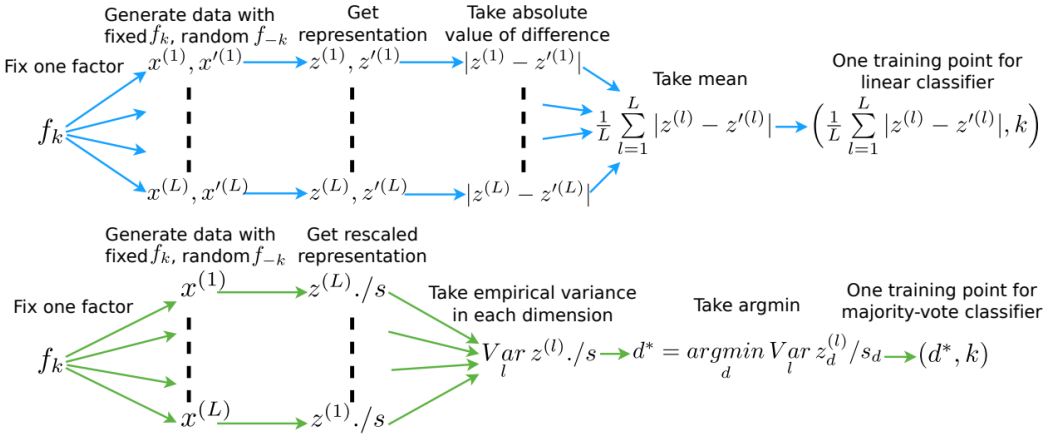

Figure 3: Comparison of the BetaVAE and FactorVAE metric [11, Fig. 2]

3.2.3 Mutual Information Gap

The disentanglement metric Mutual Information Gap (MIG), introduced by Chen et al., 2018 [12], is not based on a classifier, and instead uses the mutual information $I_n(z_j; v_k)$ between a learned latent space feature z_j and a true underlying latent variable v_k. By taking the difference between the two learned latent space features which have the highest mutual information with an underlying latent variable, high values are assigned to representations which are axis-aligned, i.e., each latent space features should contain information about only one true underlying factor and representations which are compact, i.e., only one latent space feature should contain information about a true underlying factor (this is similar to the concepts of disentanglement and completeness in the DCI metric and is illustrated in Figure 4). The MIG score can be calculated as follows:

$$MIG = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{H(v_k)} \left(I_n(z_{j^{(k)}}; v_k) - \max_{j \neq j^{(k)}} I_n(z_j; v_k) \right) \tag{1}$$

with $j^{(k)} = \text{argmax}_j \, I_n(z_j; v_k)$ and $H(v_k) = \mathbb{E}_{p(v_k)}[-\log p(v_k)]$. $H(v_k)$ describes the entropy of v_k and is used to normalize the mutual information.

3.2.4 Separated Attribute Predictability

Kumar et al., 2018 [8] criticize that for some disentanglement metrics, the assigned scores do not match the observable quality of the disentanglement and therefore propose the Separated Attribute Predictability (SAP) score to resolve this discrepancy. First a score matrix is constructed, where each row corresponds to a learned latent space feature and each column to a true underlying latent variable. The entries in this matrix indicate how well the latent space feature can be used to predict the underlying generative factor. Depending on the type of these generative factors, either the coefficient of determination R^2 or the classification accuracy is used to calculate the entries. Afterwards for each underlying factor/column the difference between the highest and second highest score is calculated. The SAP score is then obtained by taking the mean of these differences.

3.2.5 Disentanglement, Completeness and Informativeness

Eastwood et al., 2018 [14] propose a disentanglement metric which consists of three different criteria: disentanglement, completeness and informativeness (DCI). Similar to the SAP score, first a matrix is constructed in which the rows correspond to the learned latent space features c_i and the columns to the generative factors z_j. In this "matrix of relative importance R" [14, p. 2], the entries indicate how important a learned latent space feature is in predicting the generative factor and are calculated using appropriate regression methods (e.g., Lasso, Random forest).

Disentanglement measures the extent to which a learned latent space feature c_i only affects a single generative factor z_j. Eastwood et al. use the "'probability' of c_i being important for predicting z_j" [14, p. 2] $P_{ij} = R_{ij} / \sum_{k=0}^{K-1} R_{ik}$ and the entropy $H_K(P_{i.}) = -\sum_{k=0}^{K-1} P_{ik} \log_K P_{ik}$ to calculate the disentanglement score for each learned latent variable:

$$D_i = (1 - H_K(P_{i.})) \tag{2}$$

If the individual scores are weighted with $\rho_i = \sum_j R_{ij} / \sum_{ij} R_{ij}$ the total disentanglement score can be obtained by:

$$D = \sum_i \rho_i D_i \tag{3}$$

Completeness is the counterpart of disentanglement and indicates how well a generative factor z_j is influenced only by single latent variable c_i and can be calculated in the following way:

$$C_j = (1 - H_D(\tilde{P}_{.j})) \tag{4}$$

with the entropy $H_D(\tilde{P}_{.j}) = -\sum_{d=0}^{D-1} \tilde{P}_{dj} \log_D \tilde{P}_{ij}$.

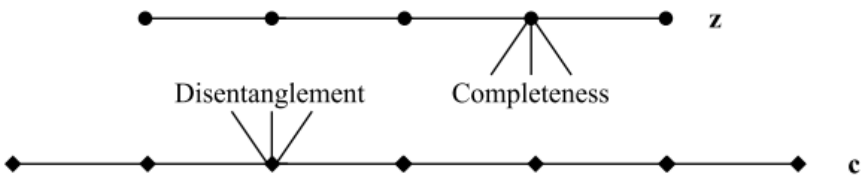

Figure 4: Comparison of disentanglement and completeness [14, Fig. 1]

Figure 4 illustrates the concepts of disentanglement and completeness. Ideally, there is a 1:1 relationship between c_i and z_j, i.e., only one learned latent space feature contains information about exactly one underlying factor.

Informativeness is the "amount of information that a representation captures about the underlying factors of variation" [14, p. 3] and is measured with an error function $E(z_j; \hat{z}_j)$, where $\hat{z}_j = f_j(\mathbf{c})$ is the prediction of the learned latent space features.

3.3 Datasets

As mentioned earlier, a prerequisite for evaluating the models with disentanglement metrics is that the true underlying generative factors of the data are known. This severely limits the number of datasets that can be used and restricts them to synthetic ones which were designed for this very purpose. The dSprites dataset [6] is probably the most commonly used dataset to study the disentanglement properties of VAEs. It consists of 5 generative factors/latent variables that change the attributes of a white object on a black background. The object can be a square, an ellipse or a heart, the x and y position of the object can be changed, as well as its scale and orientation. Besides the dSprites dataset, the Color-dSprites, Noisy-dSprites, Scream-dSprites, smallNORB [15] and Cars3D dataset will be used for the comparison.

4 Variational Autoencoders

VAEs are a special type of autoencoders, which will be discussed first to understand the basic principles.

4.1 Autoencoders

The goal of an autoencoder is to learn an identity mapping from the input to the output. In her lecture, Soleimany [2] explains that the high-dimensional input data is first fed through an encoder network to get a low-dimensional latent space representation and is afterwards given to a decoder network to reconstruct the original high-dimensional input data. The smaller the dimensionality of the latent space representation, i.e., the smaller the bottleneck, the more the input data is compressed. The underlying idea of the autoencoder is therefore not to reconstruct the input, but to learn meaningful latent space features. The loss which is used to train the neural network via backpropagation is the squared difference between the input data $\mathbf{x}$ and the reconstructed output data $\hat{\mathbf{x}}$:

$$L(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{5}$$

However, autoencoders have a problem. In his article "Intuitively Understanding Variational Autoencoders" Shafkat [16] illustrates, that their deterministic nature leads them to learn a sparse latent space, in which it is often not possible to interpolate between the different training images and generate meaningful new ones. Figure 5 shows a two-dimensional representation of the MNIST latent space learned by an autoencoder. Interpolating for example between a 1 and a 7 would not produce a meaningful result because the autoencoder just memorized the training data but learned little about its underlying structure.

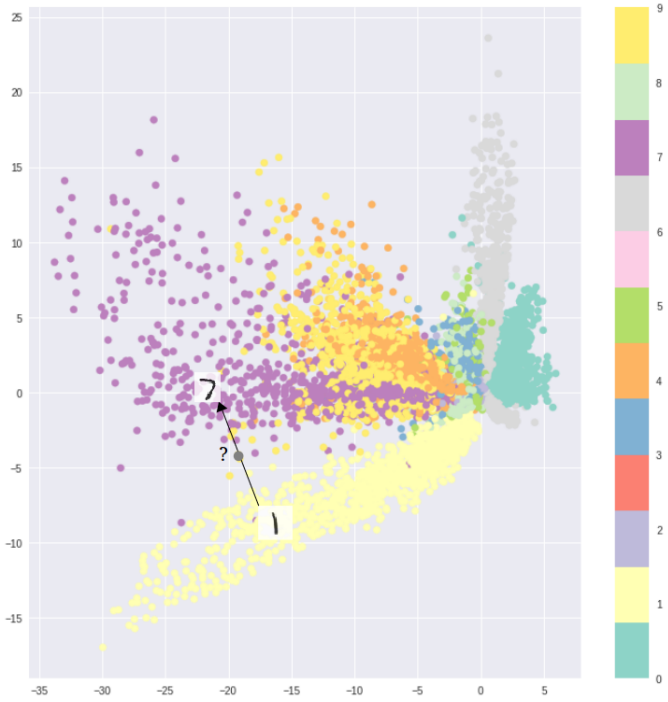

Figure 5: Visualization of the encodings from a 2D latent space learned by an autoencoder on the MNIST dataset [16]

4.2 Variational Autoencoders

To solve this problem Kingma et al., 2014 [17] introduced the Variational Autoencoder. The key idea is to replace the deterministic latent space representation $\mathbf{z}$ with a stochastic sampling process

from the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, which store the mean and standard deviation of a Gaussian probability distribution for each of the learned latent space variables [2].

In order to train the VAE using backpropagation both the loss function and the sampling operation have to be adjusted. To account for the probabilistic aspect, the loss function will be extended by a regularization term, the Kullback-Leibler (KL) divergence, which measures the difference between the learned probability distribution and a chosen prior probability distribution [2]. This leads to the following loss function [17]:

$$L_{VAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{6}$$

By forcing the learned probability distribution to be close to the chosen prior probability distribution, which is typically an isotropic Gaussian distribution with $\boldsymbol{\mu}^{(i)} = 0$ and $\boldsymbol{\sigma}^{(i)} = 1$, the VAE is encouraged to learn a dense latent space representation [2]. Figure 6 from Shafkat [16] shows that the different clusters are now located close to each other around the origin of the latent space. Generating new samples or interpolating between existing ones will now yield meaningful results.

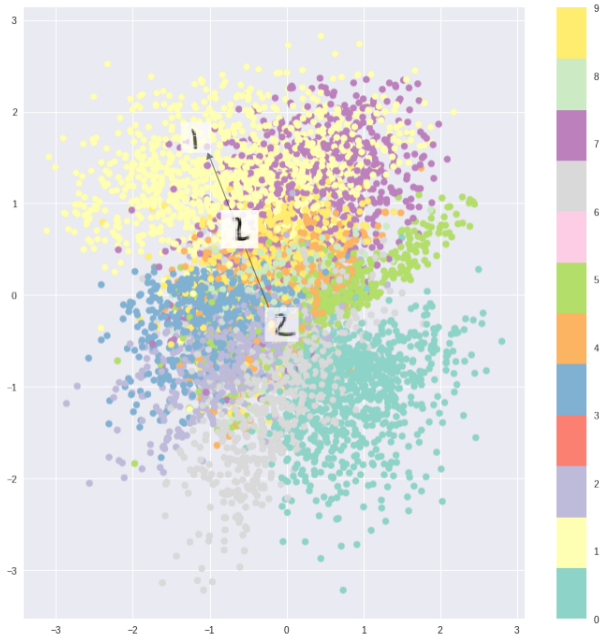

Figure 6: Visualization of the encodings from a 2D latent space learned by a VAE on the MNIST dataset [16]

Soleinmany [2] explains that the probabilistic aspect of VAEs requires an additional modification to their architecture so that they can be trained using backpropagation. For gradients to pass through the neural network, deterministic layers are needed. The stochastic sampling process in the original architecture of the VAE thus poses a problem. Kingma et al., 2014 [17] introduced the reparameterization trick to solve it: instead of drawing $\mathbf{z}$ from the probability distribution $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, $\mathbf{z}$ is instead calculated as the sum of the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, with $\boldsymbol{\sigma}$ being multiplied by $\boldsymbol{\epsilon}$ which is drawn from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. More formally:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} * \boldsymbol{\epsilon} \tag{7}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Now the stochastic sampling process only influences $\boldsymbol{\epsilon}$, which is not needed for training and as a result, gradients can simply be passed through the network and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can be trained.

4.3 Constrained Variational Autoencoders

As seen in the previous paragraph, VAEs learn an explicit representation of the latent space. If this latent space is disentangled, the individual variables/dimensions can be interpreted by humans and thus for example allow the output to be specifically modified. This puts VAEs at the center when it comes to learning disentangled representations. In recent years, many modifications to the loss function (also referred to as objective function) of the standard VAE have been proposed with the goal of encouraging the VAE to learn a disentangled latent space representation [8]–[12]. Most of them introduce additional constraints on the bottleneck capacity, which is why they will be referred to as constrained Variational Autoencoders in this seminar paper. In the following, six of them will be explained and afterwards compared regarding their ability to learn disentangled latent space features.

4.3.1 Beta-VAE

Higgins et al., 2017 [9] proposed a slight adjustment to the standard VAE by introducing the hyperparameter β which is multiplied with the regularization term in the loss function. This results in the following loss function for the β-VAE:

$$L_{\beta-VAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta\, D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{8}$$

Choosing $\beta = 1$ results in the classical VAE, but setting $\beta > 1$ forces the β-VAE to learn a more disentangled representation in order to minimize the loss function. β is used as a "regularisation coefficient that constrains the capacity of the latent information channel $\mathbf{z}$ and puts implicit independence pressure on the learnt posterior due to the isotropic nature of the Gaussian prior $p(\mathbf{z})$." [9, p. 5] The larger β is, the more importance is placed on learning a disentangled latent space, but this can come at the expense of the reconstruction quality. Therefore, when choosing the hyperparameter β, a compromise between these two factors must be found.

4.3.2 AnnealedVAE

Burgess et al., 2017 [10] want to address this problem with a change to the training procedure. They propose to increase the bottleneck capacity of the VAE incrementally so that the model learns the most important latent features at the beginning and then continues to learn the less important ones as the training progresses. The goal is to learn latent features that are similarly well disentangled as in the β-VAE but with better reconstruction quality. Therefore, the loss function of the β-VAE is expanded in the following way:

$$L_{AnnealedVAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \gamma\, |D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}) - C| \tag{9}$$

The parameter "C is gradually increased from zero to a value large enough to produce good quality reconstructions" [10, p. 7]. Setting $C = 0$ results in the β-VAE and the hyperparameter γ serves the same purpose as β in the β-VAE. In this seminar paper the name "AnnealedVAE" from Locatello et al., 2019 [7] will be used to refer to this model.

4.3.3 FactorVAE

Kim et al., 2018 [11] also try to address the trade-off problem between disentanglement and reconstruction quality in the β-VAE by introducing the FactorVAE. For this purpose, they modify the loss function of the standard VAE to more strongly penalize the dependency between the latent space variables. They use the concept of Total Correlation (TC) to measure the dependency. This leads to the following loss function:

$$L_{FactorVAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \gamma\, D_{KL}(q(\mathbf{z})||\overline{q}(\mathbf{z})) \tag{10}$$

with $\overline{q}(\mathbf{z}) := \prod_{j=1}^{d} q(z_j)$. Since $q(\mathbf{z})$ and $\overline{q}(\mathbf{z})$ are intractable to calculate, a neural network is used to approximate the TC and is trained together with the VAE. The hyperparameter γ determines how strongly the dependency between the latent space variables is penalized.

4.3.4 Beta-TCVAE

Independent from the work of Kim et al. on the FactorVAE, Chen et al., 2018 [12] introduced the β-TCVAE (Total Correlation Variational Autoencoder), which has the same objective but does not require a neural network to approximate the TC. They start by decomposing the KL divergence into three different terms:

$$\mathbb{E}_{p(\mathbf{x})}\Big[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\Big] = D_{KL}(q_\phi(\mathbf{z},\mathbf{x})||q(\mathbf{z})p(\mathbf{x})) + D_{KL}(q(\mathbf{z})||\prod_j q(z_j)) + \sum_j D_{KL}(q(z_j||p(z_j)) \tag{11}$$

The first term corresponds to the mutual information $I_q(\mathbf{z};\mathbf{x})$ between the latent space feature $\mathbf{z}$ and the sample $\mathbf{x}$. The second term is already known from the FactorVAE and describes the TC, which encourages the β-TCVAE to find statistically independent latent variables. Chen et al. refer to the last term as the dimension-wise KL, that forces the dimensions of the latent variables to be close to their corresponding prior. Based on this decomposition the following loss function for the β-TCVAE is derived:

$$\begin{aligned} L_{\beta-TCVAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \alpha\, I_q(\mathbf{z};\mathbf{x}) \\ - \beta\, D_{KL}(q(\mathbf{z})||\prod_j (q(z_j)) - \gamma \sum_j D_{KL}(q(z_j)||p(z_j)) \end{aligned} \tag{12}$$

Chen et al. emphasize that the "TC is the most important term in this decomposition for learning disentangled representations" [12, p. 4] and although α and γ can be changed as well, the experimental results suggest, that setting them to 1 yields the best results. Thus β is the only hyperparameter of the model, which needs be tuned. The individual decomposition terms are approximated using minibatch-weighted sampling.

The loss functions of the VAEs presented so far can be derived from the loss function of the β-TCVAE: the standard VAE can be obtained for $\alpha = \beta = \gamma = 1$, the β-VAE for $\alpha = \beta = \gamma > 1$ and the FactorVAE for $\alpha = \gamma = 1, \beta > 1$ [18].

4.3.5 DIP-VAE

Kumar et al., 2018 [8] introduce the DIP-VAE (Disentangled Inferred Prior), a model which encourages the disentanglement of the latent variables "by introducing a regularizer over the induced inferred prior" [8, p. 2] $q_\phi(\mathbf{z})$, also called the expected variational posterior.

They propose two versions of the DIP-VAE, which penalize the difference between the identity matrix and a) $Cov_{p(\mathbf{x})}[\boldsymbol{\mu}_\phi(\mathbf{x})]$ or b) $Cov_{q_\phi(\mathbf{z})}[\mathbf{z}]$ using the entry-wise squared l_2-norm. This leads to the following loss function for the DIP-VAE-I:

$$\begin{aligned} L_{DIP-VAE-I} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ - \lambda_{od} \sum_{i\neq j} \big[Cov_{p(\mathbf{x})}[\boldsymbol{\mu}_\phi(\mathbf{x})]\big]^2_{ij} - \lambda_d \sum_i (\big[Cov_{p(\mathbf{x})}[\boldsymbol{\mu}_\phi(\mathbf{x})]\big]_{ii} - 1)^2 \end{aligned} \tag{13}$$

And the loss function for the DIP-VAE-II is:

$$\begin{aligned} L_{DIP-VAE-II} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ - \lambda_{od} \sum_{i\neq j} \big[Cov_{q_\phi(\mathbf{z})}[\mathbf{z}]\big]^2_{ij} - \lambda_d \sum_i (\big[Cov_{q_\phi(\mathbf{z})}[\mathbf{z}]\big]_{ii} - 1)^2 \end{aligned} \tag{14}$$

The hyperparameters λ_{od} (off-diagonal) and λ_d (diagonal) determine how much the deviations from the identity matrix are penalized.

5 Quantitative Comparison of Constrained Variational Autoencoders

All the models presented above were developed to improve the learning of disentangled representations. Now the question arises which of these VAEs is best suited for representation learning and how the parameters and datasets affect the disentanglement properties of the models. For this purpose, a quantitative comparison of the different VAEs was conducted using different disentanglement metrics.

This comparison is based on the "disentanglement_lib" [19], a library which was created by Oliver Bachem and Francesco Locatello as part of the paper "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations" [7]. Each of the six VAE models (β-VAE, AnnealedVAE, FactorVAE, β-TCVAE, DIP-VAE-I, DIP-VAE-II) was trained on six different hyperparameters (Table 1), on six different datasets (dSprites, Color-dSprites, Noisy-dSprites, Scream-dSprites, smallNORB, Cars3D) and 50 different random seeds. The resulting 10.800 models along with their evaluation on various metrics were published together with the library to support future research and save computational resources. For this comparison, the relevant data for each model was extracted and processed. Afterwards, the evaluations on the disentanglement metrics BetaVAE score, FactorVAE score, MIG, SAP, DCI together with the reconstruction loss were aggregated according to different research questions and plotted using the Seaborn library [20] to compare the disentanglement properties of the models. Both the evaluations of the models as well as the plots were uploaded online.[1]

Table 1: Hyperparameters of the models [7, Tab. 3]

Model	**Parameter**	**Values**
β-VAE	β	[1, 2, 4, 6, 8, 16]
	c_{max}	[5, 10, 25, 50, 75, 100]
AnnealedVAE	iteration threshold	100000
	γ	1000
FactorVAE	γ	[10, 20, 30, 40, 50, 100]
DIP-VAE-I	λ_{od}	[1, 2, 5, 10, 20, 50]
	λ_d	$10\lambda_{od}$
DIP-VAE-II	λ_{od}	[1, 2, 5, 10, 20, 50]
	λ_d	λ_{od}
β-TCVAE	β	[1, 2, 4, 6, 8, 10]

5.1 Results

5.1.1 Which model is best suited for learning disentangled representations, i.e., achieves the highest scores on the disentanglement metrics?

Figure 7 shows that the β-TCVAE achieves the highest median and mean disentanglement scores, followed by the FactorVAE and β-VAE. The AnnealedVAE, DIP-VAE-I and DIP-VAE-II are outperformed and score significantly worse. The fact that the β-TCVAE and FactorVAE are at the top suggests that their shared objective function is better suited for learning disentangled representations compared to other objective functions. Both explicitly penalize the TC between the latent variables and differ only in the way the TC term is calculated. However, it can also be seen that the boxplots overlap and have long whiskers and outliers. This indicates that the aggregated factors (datasets, hyperparameters, random seeds) have a strong influence on the result. The models cannot be compared well regarding the reconstruction quality, since the absolute values of the reconstruction loss differ greatly between the different datasets.

5.1.2 How well do the models transfer to other datasets?

In Figure 8, it can be seen that the relative ranking of the models ordered by their mean disentanglement scores is similar on all datasets. Again, the β-TCVAE and FactorVAE followed by the β-VAE perform best. This suggests that the relative performance of a VAE in terms of its ability to learn disentanglement representations can be transferred to other datasets.

[1] https://github.com/jakobhartmann/ISS-Seminar

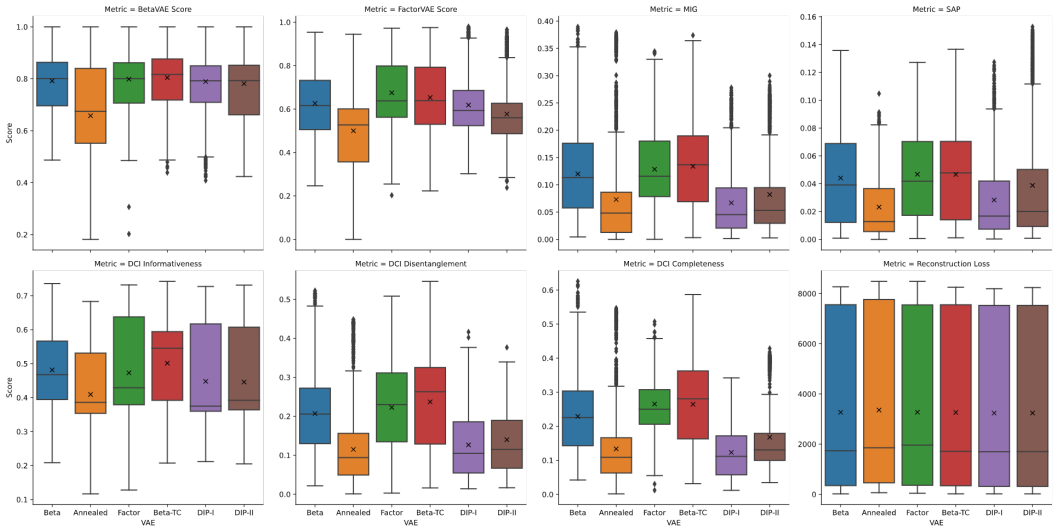

Figure 7: Evaluation of the models on different disentanglement metrics aggregated over all datasets, hyperparameters and random seeds

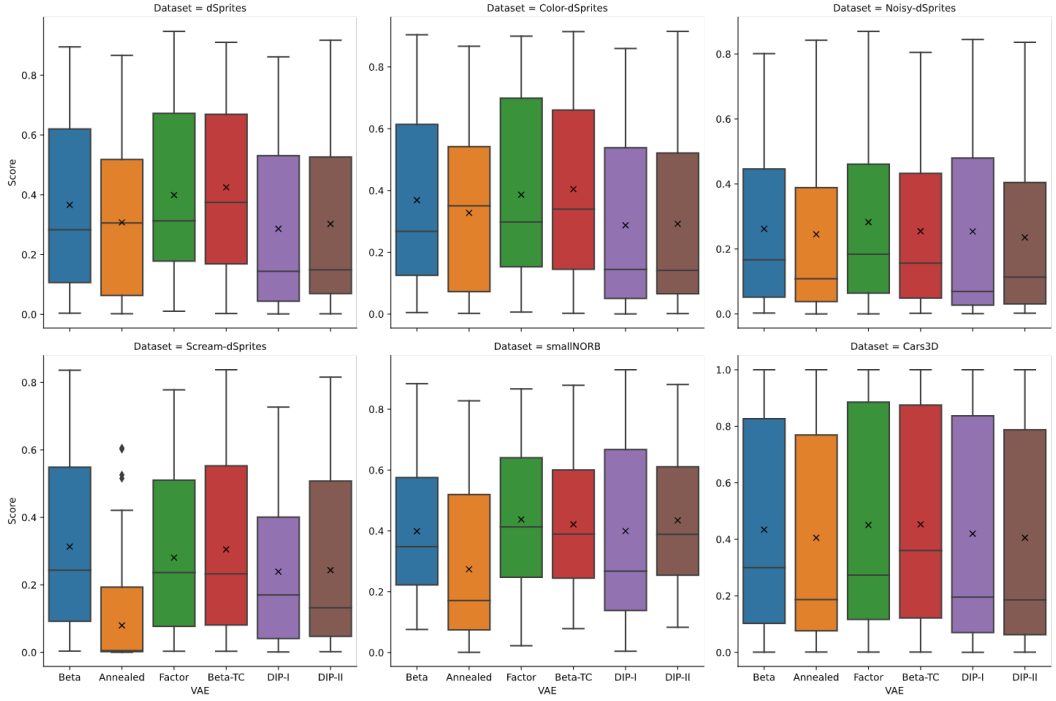

Figure 8: Evaluation of the models on different datasets aggregated over all disentanglement metrics, hyperparameters, and random seeds

5.1.3 What is the influence of hyperparameters on disentanglement scores and reconstruction loss?

Figure 9 shows that hyperparameters have a strong impact on the disentanglement scores of the models. Except for the AnnealedVAE, the hyperparameters correspond to the regularization strength, i.e., the weighting of the respective regularization term in the objective function. For the AnnealedVAE the hyperparameter corresponds to the maximum value C is increased to during training, the best results are achieved for $c_{max} = 5$. In the case of the FactorVAE and DIP-VAE-II the hyperparameters seem to only have a small influence on the disentanglement scores. The DIP-VAE-I achieves the best

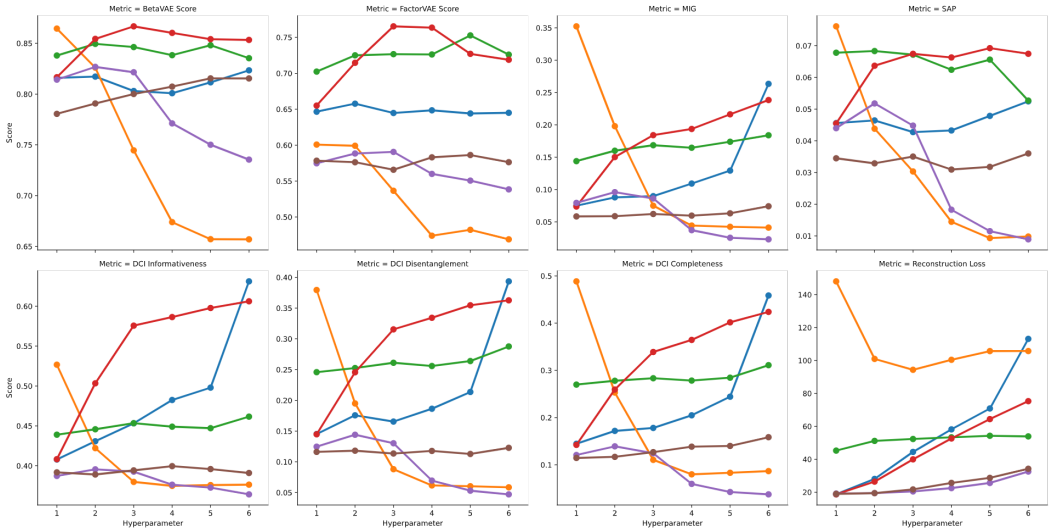

Figure 9: Evaluation of the models with their respective hyperparameters on the dSprites dataset and different disentanglement metrics averaged over all random seeds; Legend: β-VAE (blue), AnnealedVAE (orange), FactorVAE (green), β-TCVAE (red), DIP-VAE-I (purple), DIP-VAE-II (brown)

results with $\lambda_{od} = 2$. For the β-VAE and β-TCVAE, the disentanglement scores improve steadily with higher regularization strengths. However, this comes at the cost of a lower reconstruction quality. There is a positive correlation between disentanglement scores and reconstruction loss, which requires a trade-off between these two factors. Since the hyperparameters can have such a strong impact on the results, some sort of meta-learning will be required to find the optimal parameters.

5.1.4 How well do the hyperparameters of the models transfer between different datasets?

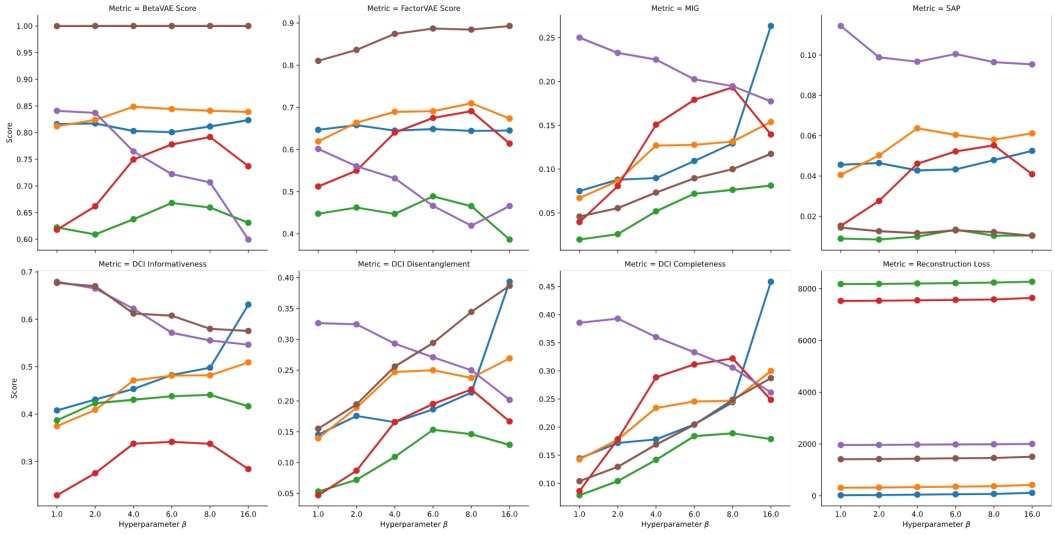

Figure 10: Evaluation of the β-VAE with different β on different datasets and disentanglement metrics averaged over all random seeds; Legend: dSprites (blue), Color-dSprites (orange), Noisy-dSprites (green), Scream-dSprites (red), smallNORB (purple), Cars3D (brown)

If good hyperparameters have been found for each model on a dataset, the question arises whether these can also be transferred to other datasets. Figure 10 indicates that the answer is no. The optimal hyperparameter for the β-VAE varies greatly between different datasets. While on the dSprites,

Color-dSprites and Cars3D dataset the best results are achieved for $\beta = 16$, the optimal β is about half that large for Noisy-dSprites and Scream-dSprites and for the smallNORB dataset it is even $\beta = 1$, i.e., the standard VAE obtains the highest scores. Although the differences are not that large for all models, a separate hyperparameter search on each dataset is necessary for all of them.

5.1.5 Do random seeds affect disentanglement scores?

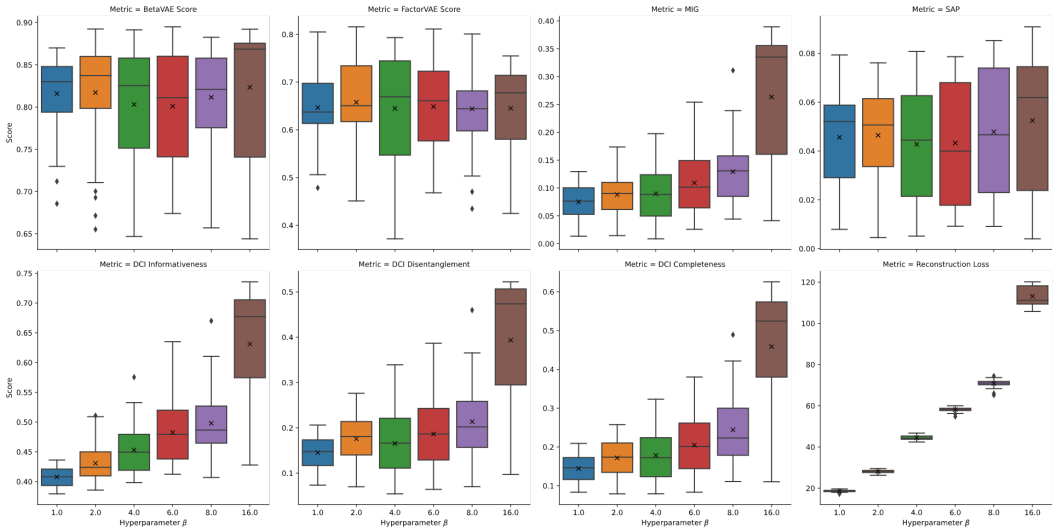

Figure 11: Evaluation of the β-VAE with different β on the dSprites dataset and different disentanglement metrics

Since a separate hyperparameter search is required for each model and dataset, it seems questionable from a resource point of view to train these models several times on different random seeds. Is this necessary? In Figure 11 each boxplot represents 50 models trained on the same hyperparameter, model and dataset but with different random seeds. The long whiskers and outliers show that the results can vary greatly from training run to training run. Locatello et al., 2019 state "that a good run with a bad hyperparameter can beat a bad run with a good hyperparameter" [7, p. 5]. To obtain a reliable result, it is therefore necessary to train the models several times in order to minimize the influence of randomness and ultimately find the best parameters.

6 Conclusion and Outlook

There are of course limitations to this comparison. Since it is entirely built on the disentanglement_lib and the pretrained models, the results are dependent on the design decisions of the developers. The hyperparameter specification of the AnnealedVAE for example is questionable. A regularization strength of $\gamma = 1000$ is exceptionally large compared to other models. It cannot be ruled out that with other hyperparameters the model performance could be increased significantly. Nevertheless, a few important conclusions can be drawn:

1. Explicitly penalizing the TC in the objective function as in the FactorVAE and β-TCVAE seems to be the best way to learn disentangled representations in VAEs. This can be observed on almost all datasets.
2. The hyperparameters in particular, but also the random seeds can have a significant impact on the disentanglement properties of the models. Therefore, it is necessary to train the models with different hyperparameters and random seeds to obtain reliable and optimal results. Meta-learning can help with this process. However, the hyperparameters do not transfer well to other datasets. Therefore, this parameter optimization has to be done separately for each dataset.

3. Although some models have been developed with the intention of aligning disentanglement and reconstruction quality, this trade-off remains an issue and requires developers to weigh these two factors on an individual basis.
4. A key issue in representation learning is the absence of a standardized method to quantitatively measure the disentanglement properties of a model. In this seminar paper, several metrics have been used to get an objective result, but in a lot of cases the metrics contradict each other in their evaluation and ranking of the models. Especially a metric for datasets where the true underlying latent factors are unknown or hierarchically ordered would help to compare existing models and develop new ones.

This seminar paper focused on constrained VAEs, which were specifically designed for learning disentangled representations. It would be interesting to extend this comparison to recently published VAEs (e.g., NVAE [21]), other architectures such as GANs (e.g., InfoGAN [22], StyleGAN [23]), Adversarial Autoencoders [24] or flow-based generative models (e.g., Glow [25]) and application areas outside of image generation in the future.

References

[1] Wikipedia contributors, *Generative model*, Wikipedia, The Free Encyclopedia, Apr. 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Generative_model&oldid=1016703868 (visited on 04/15/2021).

[2] A. Soleimany, *Deep Generative Modeling*, MIT 6.S191: Introduction to Deep Learning, Feb. 2020. [Online]. Available: https://www.youtube.com/watch?v=rZufA635dq4 (visited on 04/12/2021).

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[4] E. Tiu, *Understanding Latent Space in Machine Learning*, Medium, Feb. 2020. [Online]. Available: https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d (visited on 04/12/2021).

[5] Y. Bengio, "Learning Deep Architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009, Place: Hanover, MA, USA Publisher: Now Publishers Inc., ISSN: 1935-8237. DOI: 10.1561/2200000006.

[6] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, *dSprites: Disentanglement testing Sprites dataset*, 2017. [Online]. Available: https://github.com/deepmind/dsprites-dataset/ (visited on 04/13/2021).

[7] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, PMLR, Jun. 2019, pp. 4114–4124. [Online]. Available: http://proceedings.mlr.press/v97/locatello19a.html.

[8] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. [Online]. Available: https://openreview.net/forum?id=H1kG7GZAW.

[9] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl.

[10] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-VAE," in *NIPS Workshop on Learning Disentangled Representations*, 2017. [Online]. Available: http://arxiv.org/abs/1804.03599.

[11] H. Kim and A. Mnih, "Disentangling by Factorising," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, PMLR, Jul. 2018, pp. 2649–2658. [Online]. Available: http://proceedings.mlr.press/v80/kim18b.html.

[12] R. T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating Sources of Disentanglement in Variational Autoencoders," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/1ee3dfcd8a0645a25a35977997223d22-Paper.pdf.

[13] D. Morton, *Learning Disentangled Representations — Part 1 (simple dots)*, Medium, Jun. 2018. [Online]. Available: https://medium.com/@davidlmorton/learning-disentangled-representations-part-1-simple-dots-c5553ecc995b (visited on 04/12/2021).

[14] C. Eastwood and C. K. I. Williams, "A Framework for the Quantitative Evaluation of Disentangled Representations," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. [Online]. Available: https://openreview.net/forum?id=By-7dz-AZ.

[15] Y. LeCun, Fu Jie Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004. DOI: 10.1109/CVPR.2004.1315150.

[16] I. Shafkat, *Intuitively Understanding Variational Autoencoders*, Medium, Apr. 2018. [Online]. Available: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf (visited on 04/13/2021).

[17] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114v10.

[18] Y. Dubois, *Disentangling-vae*, GitHub. [Online]. Available: https://github.com/YannDubs/disentangling-vae (visited on 04/12/2021).

[19] O. Bachem and F. Locatello, *Disentanglement_lib*. [Online]. Available: https://github.com/google-research/disentanglement_lib (visited on 04/12/2021).

[20] M. L. Waskom, "Seaborn: Statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021, Publisher: The Open Journal. DOI: 10.21105/joss.03021.

[21] A. Vahdat and J. Kautz, "NVAE: A Deep Hierarchical Variational Autoencoder," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 19 667–19 679. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

[22] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Curran Associates Inc., 2016, pp. 2180–2188, ISBN: 978-1-5108-3881-9.

[23] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4396–4405. DOI: 10.1109/CVPR.2019.00453.

[24] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial Autoencoders," *CoRR*, vol. abs/1511.05644, 2015. [Online]. Available: http://arxiv.org/abs/1511.05644.

[25] D. P. Kingma and P. Dhariwal, "Glow: Generative Flow with Invertible 1x1 Convolutions," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.