Improved disentangling in VAE-GANs

G079 (s1650091, s1621263, s1652379)

Abstract

Recent research in probabilistic generative models based on deep neural networks has led to image generation systems of a quality previously unseen. We re-explore an algorithm first introduced by Larsen et al. (2016) that combines Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). Its aim is to combine the strength of the two approaches by informing the VAE's loss with the GAN's discriminator, thus creating a *feature*-wise loss. This helps find *disentangled* latent representations, which often better capture features of the data capable of generalizing beyond training samples (Burgess et al., 2018; Ridgeway, 2016). These result in improved sample generation. We explore the reproducibility challenges of the algorithm, compare it with VAE and GAN and augment it with β-VAE, an extension that has been shown to improve the disentanglement of latent representation in VAE. This choice involves the tuning of one key hyperparameter: we avoid the expensive heuristics proposed in the literature and show improved results on our baselines with a simple Bayesian optimization procedure with a $\approx$ 10% decrease in Frechet Inception Distance score.

1. Introduction

With the introduction of deep architectures in generative models, the task of data generation entered a new phase with unprecedented performance. However, complex data distributions such as images still remain a challenge for generative models. In an attempt to tackle this challenge, it was proposed to replace similarity metrics commonly used in generative models with a learned similarity measure (Larsen et al., 2016). The authors argued that similarity metrics such as the squared error, which is the default choice for Variational Auto Encoders (VAEs) as per Kingma & Welling (2014), are overly simplistic for image data. They do not model the characteristics of human visual perception. The proposed solution was to measure image similarity using a higher-level representation of the images. Combining a VAE and a Generative Adversarial Network (GAN) (first presented by Goodfellow et al. (2014b)) into one single architecture (Figure 2) allows the use of the GAN discriminator as a similarity measure, hence a feature-wise similarity metric. Moreover, Larsen et al. (2016) argue that the result of this loss is to find disentangled latent representations. This means that different latent units learn to encode different independent ground-truth generative factors of variation in the data. These representations have been found to improve image quality generation (Chen et al., 2016; Kim & Mnih, 2018).

Since Larsen et al. presented the VAE-GAN model, better performing architectures of both, VAEs and GANs, were introduced to the field. We focus our work on β-VAE presented by Higgins et al. (2017a), since it was successfully applied to discover factorised latent representations from image data, which was part of the problem that the VAE-GAN model above addressed. In order to improve on VAE-GAN, we then propose β-VAE-GAN. Additionally, we aim to optimize the β hyperparameter via Bayesian optimization (Snoek et al., 2012).

We first discuss related literature in Section 1.1 as well as our motivations and contributions in Section 1.2. In Section 2 we, describe the data we were using to carry out the experiments. Subsequently, we provide the background of the VAE and GAN frameworks, as well as introduce Bayesian optimization in Section 3. We then elaborate on our proposed methodology to develop an improved model in Section 4 and present results of the experiments in Section 5. Finally, we discuss our findings in Section 6.

1.1. Related Work

One of the ways the generative models literature has sought for better image generation is through disentangled latent representations. In our work we employ one of the earlier contributions on this line, β-VAE (Higgins et al., 2017a) (described in Section 4). Other approaches that address disentangling in deep generative models to improve image quality generation include Factor-VAE (Kim & Mnih, 2018). Similar to β-VAE, this algorithm introduces a penalty term to the standard VAE objective that directly encourages independence in the latent space. This term is in the form of Kullback–Leibler (KL) divergence between the marginal distribution of the latent codes and the product of its individual dimensions' marginals. They show empirically that this achieves similar image generation quality as β-VAE but without degrading reconstructions as much. This approach is promising but adds computational overhead. Furthermore, sampling from the required distributions for the KL divergence is difficult (Kim & Mnih, 2018). Since VAE-GAN already adds computational overhead to the standard VAE, we chose to proceed with β-VAE instead.

There is also work on GAN variants on this line: InfoGAN modifies the learning objective of GAN with the addition of a term that encourages high mutual information between the distribution of latent codes and that of the generator (Chen et al., 2016). Drawbacks of this approach, as discussed for example in the work of Higgins et al. (2017b); Burgess et al. (2018) are reduced sample diversity and training instability. While VAE-GAN mostly avoids the former, unfortunately it cannot escape the latter (being a GAN hybrid). Finally, relatively recent preliminary work on more advanced hybrids that try to combine VAE and GAN exist. These also implement similar approaches to construct a feature-based loss (or similarity metric) (Rosca et al., 2017).

1.2. Motivations and Contributions

One motivation to choose Larsen et al.'s paper is that it showcased an interesting variation on traditional VAEs. The authors attempt to learn a similarity metric between fake/reconstructed and real data through the aid of a GAN's discriminator. Thereby they enabled the network to produce higher quality fake images, if trained appropriately. We found that several other improvements on GANs/VAEs have been discovered since the paper's release which we could experiment with and present our findings for. The new combinations with the aforementioned β-VAE as well as the application of Bayesian optimization present a novel, unexplored space. Research on GANs as well as VAEs heavily increased over the past years and supplying further insights on the compatibility between these findings poses a sufficient vector of interest to our project.
Our aims are to address the following:

1. What are the main challenges in the reproducibility of the algorithm presented by Larsen et al. (2016) and its results? How much hyperparameter tuning is needed?

2. Is there significant benefit, in terms of visual fidelity of generated samples and reconstructions, to use VAE-GAN instead of standard VAE or GAN?

3. Since one of the main contributions of VAE-GAN is its ability to find disentangled latent representations of data, how can we improve on this with more recent advances that address this problem?

4. Can we optimize crucial hyperparameters efficiently with Bayesian optimization?

In order to address (3), we modify the VAE-GAN with *β-VAE* (Higgins et al., 2017a). We describe what is meant by disentangled representations and the method of β-VAE in the remainder of this report.

2. Dataset

We use aligned and cropped face images from the CelebA dataset to train and test our models (Liu et al., 2015). The dataset contains 202,599 images of faces as well as 5 landmark locations and 40 binary attributes annotations per image. These attributes encode content such as *wavy hair*, *mustache* and *eyeglasses*. The faces in the dataset are aligned using similarity transformation according to the two eye locations and are of size 218×178. We discard all attributes for unsupervised training of our models and by choosing to follow the design choices of Larsen et al., we resize the images to 64×64 for training and testing. This reduces the representation size and simplifies convolution operations due to the convenient equivalence in height and width. Several sample images are presented in Figure 1. Ultimately, we chose to reserve 162,770 images for training, 19,866 for validation, and 19,963 for testing. This results in an approximate 80/10/10 split of the data and is in line with the official data splits provided by Liu et al. (2015).

Figure 1. Samples from CelebA dataset in original image size 218×178. For training, 64 × 64 (down-scaled) images are used.

3. Background

In this section we provide further background on VAE and GANs, two flexible frameworks that can be used to design and train generative models.

3.1. Variational Autoencoders

In essence, a VAE is equivalent to non-linear Factor Analysis, a classic latent variable probabilistic model, using deep neural networks as nonlinear functions. (Fruchter, 1954). Given a parameterized probabilistic model $p_{\theta}(\mathbf{x}, \mathbf{z})$, where $\mathbf{z}$ denotes the latent variables and $\mathbf{x}$ the data, we usually want to infer something about the latents having observed data. This can be done with the rules of probability and an appropriate prior belief on the latents:

$$p_{\theta}(\mathbf{z} \mid \mathbf{x}) = \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{x})} = \frac{p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z})}{p_{\theta}(\mathbf{x})} \tag{1}$$

Here, the generative model $p_{\theta}(\mathbf{x}, \mathbf{z})$ is decomposed into the product of the prior, $p_{\theta}(\mathbf{z})$, and a stochastic decoder $p_{\theta}(\mathbf{x}|\mathbf{z})$. While the choice of a prior may feel awkward at first, we will see that it can be used as a powerful tool (for example, to encourage disentangled representations, as explained in Section 4). Since computing $p_{\theta}(\mathbf{z} \mid \mathbf{x})$ is an intractable integral over the complete parameter space, it is approximated with a parameterized stochastic encoder $q_{\phi}(\mathbf{z}|\mathbf{x})$. It is possible to jointly train the decoder and encoder paramters θ and ϕ by maximizing a lower bound to the marginal likelihood of the data $p_{\theta}(\mathbf{x})$. This is referred to as *Evidence Lower Bound* (ELBO $\mathcal{L}_{\theta,\phi}(\mathbf{x})$) with respect to both parameter vectors:

$$\begin{aligned}
\mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left[\log p_{\theta}(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left(q_{\phi}(\mathbf{z}|\mathbf{x}) \| p_{\theta}(\mathbf{z})\right) && (2)\\
&= \log p_{\theta}(\mathbf{x}) - D_{KL}\left(q_{\phi}(\mathbf{z}|\mathbf{x}) \| p_{\theta}(\mathbf{z}|\mathbf{x})\right) && (3)\\
&\leq \log p_{\theta}(\mathbf{x}) && (4)
\end{aligned}$$

We can easily see that (4) indeed is a lower bound due to the non-negativity of the KL divergence term. It is clear from (3) that maximizing this (or minimizing the negative, if treated as a loss) will also minimize the inclusive KL divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$. Thereby, making the encoder more accurate. The two additive terms of the ELBO are renamed for our purposes as:

$$\text{ELBO} = \mathcal{L}^{\text{pixel}}_{\text{like}} + \mathcal{L}_{\text{prior}} \tag{5}$$

with $\mathcal{L}^{\text{pixel}}_{\text{like}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right]$ (interpretable as the data reconstruction loss) and $\mathcal{L}_{\text{prior}} = D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})\right)$ (interpretable as the discrepancy of the posterior from the prior). Finally, note that the objective in 5 cannot be optimized directly because of the sampling from $q_\phi(\mathbf{z}|\mathbf{x})$. Therefore, the *reparameterization trick* (Kingma & Welling, 2014) is used to express $\mathbf{z}$ as a deterministic variable by introducing an auxiliary random variable with a fixed distribution.

3.2. Generative Adversarial Networks (GANs)

In the GAN framework, two flexible parametric models (e.g. neural networks) are jointly trained via a mini-max game (Goodfellow et al., 2014b). The *generator* attempts to generate samples that are close to the training data from the latent variables, while the *discriminator* seeks to correctly identify whether a received image sample is produced by the generator or is a real datapoint. The joint objective function to optimize with respect to the parameters of both the discriminator and the generator is:

$$\mathcal{L}_{\text{GAN}} = \log(\text{Dis}(\mathbf{x})) + \log(1 - \text{Dis}(\text{Gen}(\mathbf{z}))) \tag{6}$$

Where $\text{Dis}(\cdot)$ denotes the discriminator, and $\text{Gen}(\cdot)$ the generator. In Algorithm 1 of Larsen et al. (2016), the VAE decoder and the GAN generator are one and the same network as illustrated in Figure 2. Moreover, as explained in Section 4, the implemented loss will have an additional term as it was found in Larsen et al. (2016) to give better empirical results.

3.3. Bayesian optimization (BO)

Hyperparameter tuning is often a crucial factor to achieve optimal performance in machine learning models, especially with deep neural networks. However, the process of finding hyperparameters that generalize well on unseen data is both expensive and difficult. The most popular approach is cross-validation or k-fold cross-validation. The downside of these methods is that they are computationally expensive and decrease the amount of data available for training. In VAE-GAN, as we will discuss in detail in Section 4, hyperparameter tuning proves to be particularly crucial, mostly due to the GAN component of the architecture. A principled approach to hyperparameter tuning that uses all data (without the need for a validation set) is *Bayesian optimization* (BO).

It is a general approach to optimizing unknown, expensive functions. In our context, the underlying expensive function that we want to optimize is the quality of our experiment, using the *Fréchet Inception Distance* (FID) value calculated for the generated samples as introduced by Heusel et al. (2017) (further elaborated in Section 5). The main idea of the approach follows the general Bayesian paradigm: we should make use of *all* available information to conduct inference. This results in defining a prior over the function that we believe represents the mapping between hyperparameters and true "experiment score". A typical assumption is that of "smoothness": we don't expect the quality of the experiment to change abruptly with only a small change in hyperparameters. This can be achieved with a Gaussian Process (GP) prior with an radial basis function kernel, which is what we will use in our experiments (Section 5). The prior belief on the function values can then be updated by sampling from the expensive function, i.e. carrying out an experiment with a specific hyperparameter setting.

Since the goal of Bayesian optimization is both testing hyperparameters that seem promising (i.e. yield high quality experiments) and learn something new (try values in unexplored regions), *acquisition functions* are developed. We model the acquisition function $a(\mathbf{x})$ to indicate the expected desirability of evaluating f at a specific value x. We then optimize $a(\mathbf{x})$ to select the optimal value of x for the next evaluation of f. For our experiments we opted to use *expected improvement* (EI) (Snoek et al., 2012) as our acquisition function.

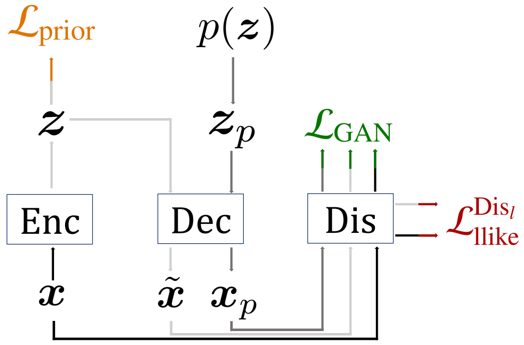


Figure 2. VAE-GAN architecture. Illustration based on the architecture presented by Larsen et al. (2016)

4. Methodology

Here we describe the algorithm to train an unsupervised VAE-GAN model (Larsen et al., 2016). Moreover, we propose an improved algorithm using β-VAE (Higgins et al., 2017a). We discuss how this adds a single hyperparameter β to control the degree of disentanglement of the latent representation.

4.1. VAE-GAN

VAE-GAN can be thought of a modification to a basic VAE where the decoder is not only encouraged to reconstruct real images, but also takes into account the "opinion" of a GAN's discriminator. This teacher network is trained normally as in GANs. Therefore, the architecture consists of three networks with distinct trainable parameters: an encoder, a decoder (which also plays the role of a GAN generator) and a discriminator as illustrated in Figure 2. These three networks are trained with distinct losses which, however, share some components. The three basic components of the loss are the following:

$$\mathcal{L}_{\text{prior}} = D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right) \tag{7}$$

$$\mathcal{L}_{\text{llike}}^{\text{Dis}_l} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\text{Dis}_l(\mathbf{x} \mid \mathbf{z}))\right] \tag{8}$$

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = \log(\text{Dis}(\mathbf{x})) + \log(1 - \text{Dis}(\text{Gen}(\mathbf{z})))+ \\ \log(1 - \text{Dis}(\text{Gen}(\mathbf{z_p})))\end{aligned} \tag{9}$$

The first component (7) calculates the discrepancy between the inferred latents $\mathbf{z}$ and the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ (standard used by the VAE). Note that this prior is appropriate in our context as such isotropic (spherical) Gaussian distributions encourage conditionally independent latent factors. Component (8) is the standard reconstruction loss of a VAE, with an important difference: instead of comparing a real input with a reconstructed one, the corresponding discriminator's *internal representations*, or $l - th$ features, are compared. This is the "learned", or "feature-based" similarity metric as it depends directly on a trainable discriminator.

This is a key contribution of VAE-GAN (Larsen et al., 2016), contrasted to the standard VAE similarity, which is pixel-based. Notational clarification: the operator $Dis(\cdot)$ defines the output of a full forward pass through the discriminator (which is a scalar probability), whereas the operator $Dis_l(\cdot)$ refers to the discriminator's l-th layer representation. Here l is chosen arbitrarily, however it is sensible to choose a layer close to the final logistic sigmoid as here the higher level features contribute to the resulting vector. We used the next to last layer as our l. Finally, the third component (9) is the standard GAN loss with an added middle term which uses a reconstructed input. Here, $\mathbf{z}_p$ denotes a prior sample. This addition was found by Larsen et al. (2016) to perform better in practice.

The density $p(\text{Dis}_l(\mathbf{x} \mid \mathbf{z}))$ in the reconstruction loss $\mathcal{L}_{\text{llike}}^{\text{Dis}_l}$ is defined as a Normal distribution on $\text{Dis}_l(\mathbf{x})$ with mean $\text{Dis}_l(\tilde{\mathbf{x}})$ and identity covariance:

$$p(\text{Dis}_l(\mathbf{x} \mid \mathbf{z})) = \mathcal{N}(\text{Dis}_l(\mathbf{x}) \mid \text{Dis}_l(\tilde{\mathbf{x}}), \mathbf{I}) \tag{10}$$

Where $\tilde{\mathbf{x}}$ is a *reconstructed* sample. Note that because of this Gaussian observation model on the $l - th$ features, we can calculate explicitly the reconstruction component:

$$\mathcal{L}_{\text{llike}}^{\text{Dis}_l} = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\text{Dis}_l(\mathbf{x} \mid \mathbf{z}))\right] \tag{11}$$

$$\propto -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\frac{1}{2}(\text{Dis}_l(\mathbf{x}) - \text{Dis}_l(\tilde{\mathbf{x}}))^\top(\text{Dis}_l(\mathbf{x}) - \text{Dis}_l(\tilde{\mathbf{x}}))\right] \tag{12}$$

$$\propto \approx -\frac{1}{2}\sum_{i=0}^{N}(\text{Dis}_l(\mathbf{x}) - \text{Dis}_l(\tilde{\mathbf{x}}^i))^\top(\text{Dis}_l(\mathbf{x}) - \text{Dis}_l(\tilde{\mathbf{x}}^i)) \tag{13}$$

Where the expectation is approximated with Monte Carlo samples from the probabilistic encoder $\mathbf{z}^i \sim q_\phi(\mathbf{z}|\mathbf{x})$, and $\tilde{\mathbf{x}}^i$ denotes that the reconstructed $\tilde{\mathbf{x}}$ was generated using sample $\mathbf{z}^i$. We can ignore the normalizing constant of the Gaussian as it is an additive constant term to the loss. Here, we do not expand in detail $\mathcal{L}_{\text{prior}}$ and $\mathcal{L}_{\text{GAN}}$, as they are standard terms and their computation details can be found respectively in Kingma & Welling (2014); Goodfellow et al. (2014a).

Given the three main components, the losses for the networks are as follows:

$$\mathcal{L}_{\text{encoder}} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l} \tag{14}$$

$$\mathcal{L}_{\text{decoder}} = \gamma \cdot \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}} \tag{15}$$

$$\mathcal{L}_{\text{discriminator}} = \mathcal{L}_{\text{GAN}} \tag{16}$$

Where γ is a tunable hyperparameter that weighs the ability to reconstruct vs that of fooling the discriminator. This weighting is interpreted by Larsen et al. (2016) as balancing content vs style of the generated samples respectively. Unfortunately Larsen et al. (2016) do not provide used values or guidance on suitable values for this parameter.

The training algorithm is fully described in Algorithm 1. As a summary, we get terms $\tilde{\mathbf{X}}, \mathbf{X}_p$ by encoding and decoding a real input $\mathbf{X}$ and just decoding some noise $\mathbf{Z}_p \sim \mathcal{N}(0, 1)$, respectively. Then, we calculate the three components of the loss and update parameters of the three networks separately, based on the appropriate combinations of losses as given by 15,16 and 17.

4.2. Better disentangling with β-VAE

In order to improve the ability of the VAE-GAN to generate latent representation that are associated with real, distinct generative factors of the data (e.g. skin colour, having a mustache, etc.), we modify the framework to incorporate the changes introduced by the β-VAE. This improvement was proposed by Higgins et al. (2017a): the main idea is that a constraint on the original VAE ELBO is imposed so that statistical independencies in the latent vector are encouraged. The aim is to obtain so called *disentangled* latent representations of the data. These are defined as representations where single latent units are sensitive to changes in single, ground-truth generative factors, while being relatively invariant to changes in other factors (Higgins et al., 2017a; Burgess et al., 2018). It has been argued that these kinds of representations may be helpful for generalization and intepretability of latent space (Bengio et al., 2013; Ridgeway, 2016). To achieve this goal, the following modified "β-ELBO", $\mathcal{L}_{\theta,\phi}^{\beta}(\mathbf{x})$ is used:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}^{\beta}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \\ - \beta \cdot D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\right)\end{aligned} \tag{17}$$

The effect of having $\beta > 1$ (since $\beta = 1$ reduces to VAE) is to encourage disentangling properties in the inferred posterior latent approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. The β constraint tries to match it to a prior $p_\theta(\mathbf{z})$ that can both control the capacity of the latent channel and encourage disentanglement. This is usually achieved with an isotropic unit variance Gaussian prior. However, too large a value of β can enforce too

Algorithm 1. Training algorithm for VAE-GAN

```
$\boldsymbol{\theta}_{\text{Enc}}, \boldsymbol{\theta}_{\text{Dec}}, \boldsymbol{\theta}_{\text{Dis}} \leftarrow$ initialize network parameters
**repeat**
1. $\boldsymbol{X} \leftarrow$ random mini-batch from dataset
2. $\boldsymbol{Z} \leftarrow \text{Enc}(\boldsymbol{X})$
3. $\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(q(\mathbf{Z}|\boldsymbol{X})\|p(\mathbf{Z}))$
   # Note here the use of $Dis_l$
5. $\mathcal{L}_{\text{llike}}^{\text{Dis}_l} \leftarrow -\mathbb{E}_{q(\mathbf{Z}|\boldsymbol{X})}\left[p\left(\text{Dis}_l(\boldsymbol{X})|\mathbf{Z}\right)\right]$
   # This is the reconstructed real input
4. $\tilde{\boldsymbol{X}} \leftarrow \text{Dec}(\boldsymbol{Z})$
6. $\boldsymbol{Z}_p \leftarrow$ samples from prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
   # This is a fake, newly generated input
7. $\boldsymbol{X}_p \leftarrow \text{Dec}\left(\boldsymbol{Z}_p\right)$
8. $\mathcal{L}_{\text{GAN}} \leftarrow \log(\text{Dis}(\boldsymbol{X})) + \log(1 - \text{Dis}(\tilde{\boldsymbol{X}})) + \log\left(1 - \text{Dis}\left(\boldsymbol{X}_p\right)\right)$
   # Update parameters according to gradients
9. $\boldsymbol{\theta}_{\text{Enc}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Enc}}} \overbrace{\left(\mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l}\right)}^{\mathcal{L}_{encoder}}$
   # Hyperparameter $\gamma$ weighs content vs style
10. $\boldsymbol{\theta}_{\text{Dec}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Dec}}} \overbrace{\left(\gamma \cdot \mathcal{L}_{\text{llike}}^{\text{Dis}_l} - \mathcal{L}_{\text{GAN}}\right)}^{\mathcal{L}_{decoder}}$
11. $\boldsymbol{\theta}_{\text{Dis}} \overset{+}{\leftarrow} -\nabla_{\boldsymbol{\theta}_{\text{Dis}}} \overbrace{\mathcal{L}_{\text{GAN}}}^{\mathcal{L}_{discriminator}}$
**until** stopping criterion
```

much similarity to the prior and severely limit quality of reconstructions.

Therefore, tuning is required to find an appropriate value of β that is neither too low nor too high. Higgins et al. (2017a) suggest that according to empirical evidence, the optimal value is dependent on both the dimensionality of the latent representation M and that of the input N, defining a "normalized" beta as $\beta_{\text{norm}} = \frac{\beta M}{N}$. We try to match similar values of β_{norm} as per Higgins et al. (2017a). They find that for values of $\beta_{norm} \geq 2$ (approximately), visual reconstruction quality degrades significantly. Therefore, we will choose values of β for our experiments that correspond to $\beta_{norm} < 2$, given our latent space size $M = 128$ and input size $N = 3 \cdot 64 \cdot 64$. Further, note that as per Higgins et al. (2017a) it is found that there is a natural "inverted U" relationship between the latent space dimension and β. When β is too low or too high, the model learns an *entangled* (opposed to disentangled) latent representation due to too little/too high capacity in the latent bottleneck. The β-VAE authors also note that reconstruction quality is a poor indicator of disentaglement. This seems to somehow suggest that there is a trade-off between good reconstructions and good fake image generation abilities to be found by hyperparameter search on β. We propose a simple and task-agnostic hyperparameter tuning method based on Bayesian optimization which we further elaborate in Section 5.4.

5. Experiments

While investigating our research we encountered several challenges in reproducing the original VAE-GAN algorithm for sample generation and reconstruction. Larsen et al. (2016) do not provide guidance on hyperparameters such as: networks weight initialization, latent vector dimension, and most importantly the content vs style weight γ. Moreover, GANs are well known to suffer severe optimization issues (Arjovsky & Bottou, 2017; Salimans et al., 2016): we discuss the issues we encountered in Section 5.5, as well as the extent to which we were able to address our research questions with our results. In Sections 5.1, 5.2 and 5.3 we discuss how we evaluated our models and present the results of each experiment.

5.1. Experiments evaluation method

Besides evaluating generated samples visually we also evaluate our reconstruction quality numerically using the FID metric (Heusel et al., 2017). This method takes as input a set of generated samples and returns a positive score. The lower the score, the closer the samples resemble the quality and style from the reference dataset (celebA in this case). For reference, we calculated the FID between a random sample of 50,000 images from CelebA and the whole dataset, resulting in a score of ≈ 2. Throughout our experiments we will compare the obtained values with this ground truth. The FID embeds the real and fake samples into the feature space of a specific layer of a pre-trained Inception network (Inception-v3 is state-of-the-art image classification network from Szegedy et al. (2016)). Assuming this representation follows a Gaussian observation model, it then calculates means and covariances for both sets of embeddings. Then, the sample quality is calculated as the Frechet distance between the two Gaussians:

$$\text{FID}\left(\mathbf{x}_p, \mathbf{x}_q\right) = \left\|\boldsymbol{\mu}_{x_p} - \boldsymbol{\mu}_{x_q}\right\|_2^2 + \text{Tr}\left(\boldsymbol{\Sigma}_{x_p} + \boldsymbol{\Sigma}_{x_q} - 2\left(\boldsymbol{\Sigma}_{x_p}\boldsymbol{\Sigma}_{x_q}\right)^{\frac{1}{2}}\right) \tag{18}$$

Where $\mathbf{x}_p$,$\mathbf{x}_q$ are the feature embeddings for a real and a fake sample, $\boldsymbol{\mu}_{x_p}, \boldsymbol{\Sigma}_{x_p}, \boldsymbol{\mu}_{x_q}, \boldsymbol{\Sigma}_{x_q}$ the corresponding Gaussian's parameters. The authors of VAE-GAN (Larsen et al., 2016) evaluate sample quality with a different method. Their approach is based on *attribute* similarity (Yan et al., 2016). The main idea here is that a good generative model should be able to produce fake images that show visual attributes which can be correctly identified by a teacher network. This teacher has been trained (through supervised learning) to identify the presence of visual attributes in generated images. The approach is described in full detail by Larsen et al. (2016); Yan et al. (2016). Since the publication of Larsen et al. (2016) in 2016, the FID score has been used as proxy for sample quality to a far larger extent due to its comparability. Literature implementing it shows strong focus on generative models (Van den Oord et al., 2016; Salimans et al., 2016; Li et al., 2017; Gulrajani et al., 2017; Huang et al., 2017; Srivastava et al., 2020). For this reason, we decided to evaluate our models with this more modern and widely used metric.

Enc	Dec	Dis
5×5 filter, 64 chann. conv. ↓, BNorm, ReLU	$8 \cdot 8 \cdot 256$, BNorm, ReLU	5×5 filter, 32 chann. conv. , ReLU
5×5 filter, 128 chann. conv. ↓, BNorm, ReLU	5×5 filter, 256 chann. conv. ↑, BNorm, ReLU	5×5 filter, 128 chann. conv. ↓, BNorm, ReLU
5×5 filter, 256 chann. conv. ↓, BNorm, ReLU	5×5 filter, 128 chann. conv. ↑, BNorm, ReLU	5×5 filter, 256 chann. conv. ↓, BNorm, ReLU
2048 fully connected, BNorm, ReLU	5×5 filter, 32 chann. conv. ↑, BNorm, ReLU	5×5 filter, 256 chann. conv. ↓, BNorm, ReLU
		1 fully connected, sigmoid

Table 1. Architectures uwased for the three neural networks involved, from left to right: Encoder, Decoder and Discriminator. These are the same as in Larsen et al. (2016). Note that the downward arrow ↓ represents that the associated layer decreases the representation size. Similarly, with the upward arrow the representation size increases. For each network, input enters at the top, output obtained at the bottom.

5.2. Experiment 1: Baselines

Our first experiment aims to replicate the VAE-GAN architecture presented and compare it with the standalone VAE and GAN architectures it is composed of. With this we address our first and second research question. We will use all three as baselines in the subsequent experiments. Throughout this encountered several difficulties with the specific settings used by Larsen et al. (2016) despite using identical network architectures (shown in Table 1) and the hyperparameters found in the alongside published code. As discussed in Section 5.5, we were required to change the optimizer from RMSProp to Adam. Despite the change, the same learning rate was used ($3 \cdot 10^{-4}$). For the latent vector we were able to find the shape used in the author's code which, alongside the other hyperparameters are presented in Table 3. As the value used for γ was not provided by Larsen et al., we chose to balance between content and style by setting $\gamma = 1$. Ultimately, in order to ease parallel computation we chose a batch size of 256. This allowed us to decrease computation time by reducing the data transfer frequency between the two NVIDIA Tesla K80 GPUs used.

In Figure 3 we report the obtained FID scores during training for all three baselines (VAE, GAN and VAE-GAN). While the VAE shows a very smooth, decreasing curve, the GAN clearly exhibits jumping and irregular progression as expected. Notably, the VAE-GAN architecture shows critically lower FID values when compared to both the VAE and GAN. It furthermore retains the smoothness exhibited by the VAE. Through visual inspection we concluded that the blurriness, typical of pictures generated by the VAE, lead to a higher FID than the GAN although recognizable face features were more present among the samples.

In Table 2 the FID scores are presented for 10 epochs. We chose this low number of epochs in order to trade-off limited availability of computation time, comparability, and to avoid mode collapse of the GAN components. While it does not allow us to provide state of the art results, we observe sufficient convergence to provide concise statements on the different baselines and our subsequent improvements. Generated samples and reconstructions can be seen in Figure 4 and 6 respectively. By inspecting these visually, we observe that VAE-GAN does provide the better samples compared to VAE and GAN, as confirmed by FID values in Table 2. In general, our samples from VAE-GAN look like VAE images but sharper, especially at the center. This is because of the component of the loss coming from the discriminator. Indeed, GAN are known to produce sharper images than VAE (Chauhan).

Baseline	FID of 1000 samples
GAN	196.00
VAE	209.00
VAE-GAN	**139.33**

Table 2. Comparison of the FID on 1000 samples for our baseline architectures.

Hyperparameter	
γ (content vs style)	1.0
Latent vector dimension	(128,1)
Optimizer	Adam
Learning rate	$3 \cdot 10^{-4}$
Weight initialization	$\mathcal{N}(0, 0.02^2)$ Conv/BNorm/FC , Biases=0

Table 3. These are hyperparameters we used across all our experiments. They were chosen from a combination of direct experimentation, reference to the VAE-GAN paper (Larsen et al., 2016) whenever possible, and official PyTorch tutorials for GAN. FC stands for Fully Connected, BNorm stands for a BatchNorm layer and Conv stands for a convolutional layer.

5.3. Experiment 2: β improvement

What the authors demonstrated with the VAE-GAN is a disentanglement of the learned latent factors. This results in higher visual fidelity generated images. Hence, we seek to improve on this using the β-VAE presented Section 4.2. By conducting training with equal hyperparameters and epoch number on our implementation of a β-VAE-GAN, we obtained the results illustrated in Table 4. As discussed in Section 4.2, a β-VAE-GAN with $\beta = 1$ is equivalent to our baseline VAE-GAN hence we conducted our experiments on values $50, 100, 150$ for initial exploration. These values were chosen as equally spaced grid due to the following calculation. From Figure 5 in Higgins et al. (2017b), we can see that reconstruction quality degrades significantly for $\beta_{norm} \geq 2$ approximately (β_{norm} is defined in Section 4.2). The values of $50, 100, 150$ correspond to calculated $\beta_{norm} = 0.52, 1.04, 1.56$, so that they cover a wide range of possibilities for β_{norm} while being lower than the threshold

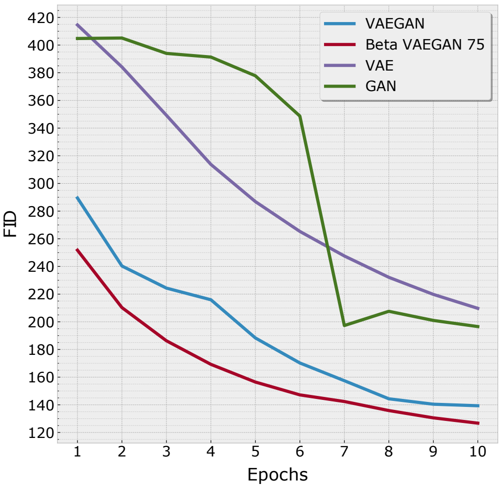


Figure 3. FID scores of VAE, GAN, VAE-GAN, and β-VAE-GAN with $\beta = 75$ for 10 epochs of training. It is likely that training for more epochs would decrease FID scores for all models, as they did not seem to have completely converged.

value devised by Higgins et al..
The resulting images showed improvements over our baseline for some values of β. However, for careful tuning of β we use Bayesian optimization, described in the next Section. FID scores for all values of β tried, including that suggested by Bayesian Optimization, are shown in Table 4.

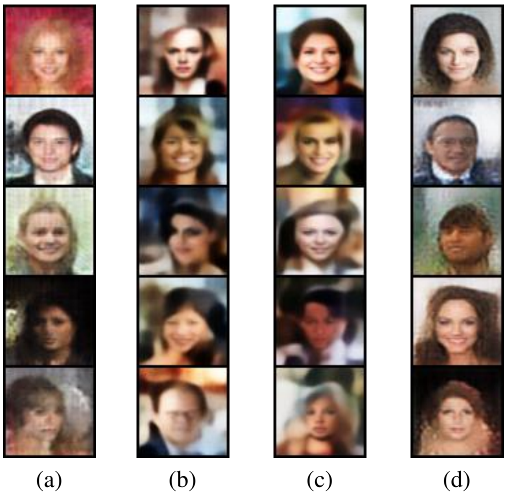


Figure 4. Generated samples for: (a) GAN, (b) VAE, (c) VAE-GAN, (d) β-VAE-GAN (the best model, with $\beta = 75$). All training conducted 10 epochs. Notice the typical blurriness of VAE sample. The VAE-GAN model only seems to slightly improve on the VAE, also removing some blurriness at the center. However, β-VAE clearly improves overall quality, although showing some of the typical defects of GAN.

5.4. Experiment 3: Bayesian Optimization on β

We use a Gaussian Process with Radial Basis Function kernel for Bayesian Optimization (BO). The expensive function to optimize is the mapping from values of β to FID of generated samples (to be minimized). Inputs to the GP are thus one-dimensional values of β, and the process is fit to our observed FID values. Therefore, the kernel function for two different values of β is:

$$k(\beta_1, \beta_2) = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{(\beta_1 - \beta_2)^2}{l^2}\right) \tag{19}$$

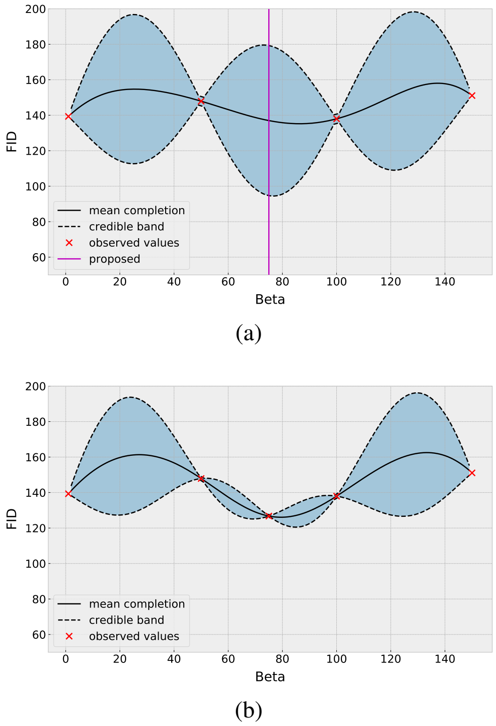
Where we used $\sigma_f = 50$ (function variance) and $l = 30$ (lengthscale). As mentioned in the background, BO allows us to do hyperparameter tuning using all data available and without a need for a validation test. Furthermore, a non trivial advantage of only optimizing one hyperparameter is that we can easily visualize the complete BO procedure. In this way, we can use our eyes beyond the suggestion of the acquisition function. The interval over which we apply BO is $[1, 150]$, for reasons explained in the previous Section. In Table 4 we highlight the lowest FID obtained by $\beta = 75$ which was the choice informed from our approach. In Figure 5 we illustrate and describe the BO procedure. As acquisition function, we used Expected Improvement (EI) , one of the simplest and most popular, especially with Gaussian Processes (Snoek et al., 2012; Shahriari et al., 2015). It is based on the idea that we would like the new candidate point to minimize the distance to the objective evaluated at its maximum. Since we do not know the location of the maximum, we replace that with the current maximum. The recommended β from this procedure, given observed pairs from Table 4 (without $\beta = 75$) was 76.32, which we rounded to 75.

Improved VAE-GAN	FID of 1000 samples
$\beta = 50$	147.92
$\beta = 75$	**126.75**
$\beta = 100$	137.98
$\beta = 150$	151.08

Table 4. Comparison of final FID on generated samples for different values of β. We believe too high values of beta $\approx\geq 150$ i.e. $\beta_{norm} \geq 2$ will not improve FID. The best value obtained by $\beta = 75$ is a $\approx 10\%$ improvement on the baseline VAE-GAN.

5.5. Discussion

The training of VAE-GAN presented several challenges. Some of these were already outlined in Larsen et al. (2016): one has to be careful to not backpropagate gradients for the feature-reconstruction loss $\mathcal{L}^{\mathrm{Dis}_l}_{\mathrm{llike}}$ on the discriminator, as this would collapse it to 0. Moreover, it was hard to initally determine the weighting of "style" vs "content" hyperparameter γ, as authors of VAE-GAN do not provide guidelines. In the end we settled for an equal weighting ($\gamma = 1$): more tuning for optimal performance would be required. Adding the GAN component on the loss, and the discriminator to the algorithm is the most significant challenge in the implementation. Indeed, during the training of the GAN baseline we could not understand the reason for bad performance,



(a)

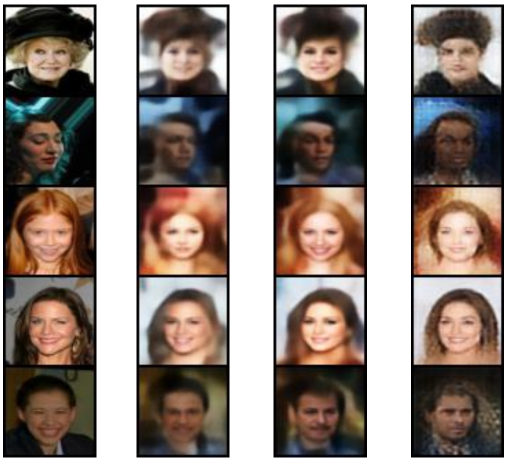
(b)

Figure 5. Bayesian optimization performed on the value of β. On the y axis, the FID value of the trained model on 1000 generated samples (lower is better). Plotted in solid black is the mean prediction of the Gaussian Process, with confidence bands representing ±2 standard deviations. In (a), we have trained models for $\beta = 1, 50, 100, 150$. The acquisition function identifies as best candidate the value of ≈ 75. This makes sense: at this location, the mean prediction is at its minimum, and the uncertainty is also high. In (b) we re-fit the GP after evaluating at $\beta = 75$ too, which indeed turned out to give the best, hence lowest, FID value. From this plot, it seems relatively unlikely to find a better value, as the lower extremes of the confidence bands are very close to the value obtained by $\beta = 75$.

and simply switching from RMSProp (used in Larsen et al. (2016)) to Adam fixed the optimization issues. Moreover, in order to stabilize GAN and VAE-GAN training, we had to use "soft" labels of 0.95 and 0.05 for 1 and 0 respectively, as well as randomly swapping them with 5% probability at each iteration. In general, we followed guidelines from ganhacks[1]; however, we decided to not change the original network architectures in VAE-GAN, in order to better understand the effects of the other changes we were making.

6. Conclusion

In this report we have shown that with some hyperparameter tuning (γ, optimizer) and GAN stabilization measures, it was possible to reproduce a comparable baseline to Larsen et al. (2016). After comparing this baseline with the results from both VAE and GAN we proposed a novel architecture, the β-VAE-GAN, to improve on the obtained results in terms of quality of generated samples. By substituting the VAE component with β-VAE, we could obtain an improved disentanglement of the latent space. The samples from this new architecture showed an improvement over the ones generated by the baseline, in terms of visual quality and FID. We found that our architecture was relatively sensitive to the newly introduced hyperparameter. In order to guarantee best results for our 10 epoch training, we conducted hyperparameter tuning through the Bayesian optimization framework. This allowed us to find a value of $\beta = 75$ for best results in terms of FID calculated on the generated samples.



(a) (b) (c) (d)

Figure 6. Reconstructions for (b) VAE, (c) VAE-GAN, (d) β-VAE-GAN ($\beta = 75$). (a) is the original image. As one can see, the reconstructions are worse for β-VAE-GAN than for all others. This is also what Higgins et al. (2017b) find for β-VAE: adding a value $\beta > 1$ to the reconstruction loss causes the reconstructions to become worse. However, other models that could be incorporated here as further work have managed to maintain's β-VAE's properties witout degrading reconstructions (Kim & Mnih, 2018).

While our results are far from state of the art in terms of visual quality, given our limited training time we were able to report guidelines on the reproducibility of VAE-GAN, showed improvements by adding β, and discovered the best setting for the introduced hyperparameter through Bayesian optimization. We conclude that VAE-GAN shows clear potential for visual improvement over VAE, and that perhaps better feature-based loss strategies could bring it to a level where it is worth the extra computation time (as one has to train three networks rather than two). It would be interesting to incorporate advancements such as FactorVAE into our work, as it claims to achieve similar results to β-VAE but without degradation of reconstruction quality (Kim & Mnih, 2018). Moreover, Wasserstein GAN has become a popular way to improve on GAN (Arjovsky et al., 2017). Notably this type of GAN has a loss that is easier to interpret and does not suffer from mode collapse as much as GAN.

[1] https://github.com/soumith/ganhacks

References

Arjovsky, Martin and Bottou, Léon. Towards principled methods for training generative adversarial networks. arxiv e-prints, art. *arXiv preprint arXiv:1701.04862*, 2017.

Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

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

Chauhan, Jaydeep T. Comparative study of gan and vae. *International Journal of Computer Applications*, 975: 8887.

Chen, Xi, Duan, Yan, Houthooft, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Fruchter, Benjamin. Introduction to factor analysis. 1954.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014a. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014b.

Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron C. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Higgins, Irina, Matthey, Loïc, Pal, Arka, Burgess, Christopher, Glorot, Xavier, Botvinick, Matthew, Mohamed, Shakir, and Lerchner, Alexander. 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*, 2017a. URL https://openreview.net/forum?id=Sy2fzU9gl.

Higgins, Irina, Matthey, Loïc, Pal, Arka, Burgess, Christopher, Glorot, Xavier, Botvinick, Matthew, Mohamed, Shakir, and Lerchner, Alexander. 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*, 2017b. URL https://openreview.net/forum?id=Sy2fzU9gl.

Huang, Xun, Li, Yixuan, Poursaeed, Omid, Hopcroft, John, and Belongie, Serge. Stacked generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5077–5086, 2017.

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

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. In Balcan, Maria Florina and Weinberger, Kilian Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/larsen16.html.

Li, Chun-Liang, Chang, Wei-Cheng, Cheng, Yu, Yang, Yiming, and Póczos, Barnabás. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ridgeway, Karl. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

Rosca, Mihaela, Lakshminarayanan, Balaji, Warde-Farley, David, and Mohamed, Shakir. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P, and De Freitas, Nando. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2951–2959. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

Srivastava, Akash, Xu, Kai, Gutmann, Michael U, and Sutton, Charles. Ratio matching mmd nets: Low dimensional projections for effective deep generative models. To appear in: 8th International Conference on Learning Representations, ICLR, 2020.

Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse, Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

Yan, Xinchen, Yang, Jimei, Sohn, Kihyuk, and Lee, Honglak. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pp. 776–791. Springer, 2016.