AVAE: Adversarial Variational Auto Encoder

Antoine Plumerault*†, Hervé Le Borgne*, Céline Hudelot†
*Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
Email: {antoine.plumerault,herve.le-borgne}@cea.fr
†MICS, Centrale-Supelec, Gif-sur-Yvette, France
Email: celine.hudelot@centralesupelec.fr

***Abstract*—Among the wide variety of image generative models, two models stand out: Variational Auto Encoders (VAE) and Generative Adversarial Networks (GAN). GANs can produce realistic images, but they suffer from mode collapse and do not provide simple ways to get the latent representation of an image. On the other hand, VAEs do not have these problems, but they often generate images less realistic than GANs. In this article, we explain that this lack of realism is partially due to a common underestimation of the natural image manifold dimensionality. To solve this issue we introduce a new framework that combines VAE and GAN in a novel and complementary way to produce an auto-encoding model that keeps VAEs properties while generating images of GAN-quality. We evaluate our approach both qualitatively and quantitatively on five image datasets.**

I. Introduction

Since the original GAN paper [1], generative models have successfully leveraged the power of deep learning to generate complex data distribution with increasing fidelity. Generative models are now used for a wide variety of tasks, including notably sample generation but also photo manipulation [2], style transfer [3], pre-processing for face recognition [4], text to image translation [5] and controlled image generation [6].

In the literature, two families of generative models stand out for image data: Variational Auto Encoders (VAE) [7] and Generative Adversarial Networks (GAN) [1], each exhibiting respective advantages and limitations. GANs usually produce more realistic images [8], [9] but they are notoriously difficult to train and suffer from mode collapse [10]. Moreover, when using GANs, there is no trivial way to get the latent representation of an image, limiting their use. In contrast, VAE models do not share these problems but the images they generate suffer from a lack of realism. It is often explained by the use of inappropriate reconstruction errors. Some previous works [11], [12], [13] have proposed solutions to solve these problems by combining or modifying these two frameworks. However, these methods exhibit a trade-off between the realism of the generated images and the fidelity of the reconstructions. In this paper, we show that GANs and VAEs can be complementary in the sense that we can derive two complementary losses from them. From this observation, we propose the AVAE model which is a VAE style model to produce samples of comparable quality as those generated by a GAN while allowing high fidelity reconstructions when used as an auto-encoder. In comparison to [11] who first introduces the idea of a combination of the two frameworks, we provide theoretical insights to show the pertinence of our approach and we address the problem of the trade-off between realism and reconstruction accuracy.

The paper is organized as follows. We begin with a reminder of GAN and VAE frameworks and explain their limitations. Then we investigate how they can be combined effectively. We thus propose an effective approach to do so, named AVAE. At last, we present a qualitative and quantitative evaluation of the performance of our model on a variety of image datasets comparing it with the state of the art. We also show that our method scales well to high resolution images.

II. Background

A. Variational Auto Encoders

VAE [7] is a framework to learn deep latent variable models. It assumes that observed data X result from random variables $z \sim p(z)$ in a latent space $\mathcal{Z}$ such that it exists a deterministic function $f : (z, \epsilon) \to x$, ϵ being a stochastic noise. The probability of observing x knowing z is estimated by a *decoder* model $p_{\theta_d} : z \mapsto p_{\theta_d}(x|z)$ parametrized by θ_d and on the contrary, the probability that z is the latent source of x is estimated by a *encoder* model $q_{\theta_e} : x \mapsto q_{\theta_e}(z|x)$ parametrized by θ_e. To estimate the parameters of the generative model of the data $X = (x^{(1)}, ..., x^{(N)})$ with N the number of observed samples, we maximize the log likelihood of the observations: $\log p_{\theta_d}\left(x^{(i)}\right) = \log \int_{\mathcal{Z}} p_{\theta_d}\left(x^{(i)} \middle| z\right) p(z) dz$. Computing $\log p_{\theta_d}\left(x^{(i)}\right)$ is nevertheless intractable in practice, thus [7] proposes to maximize a tractable lower bound, leading to the following loss to train the VAE:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}\left(\theta_e, \theta_d; x\right) = &\underbrace{\mathbb{E}_{q_{\theta_e}(z|x)}\left[-\log p_{\theta_d}\left(x|z\right)\right]}_{\mathcal{L}_{\mathcal{R}}} \\ &+ \underbrace{\text{KL}\left(q_{\theta_e}\left(z|x\right) \| \, p(z)\right)}_{\mathcal{L}_{\mathcal{P}}}\end{aligned} \quad (1)$$

with p_{θ_d} usually chosen as a Gaussian distribution $\mathcal{N}(x; \mu_{\theta_d}(z), Id)$ and KL the Kullback-Leibler divergence. Hence, the term $\mathcal{L}_{\mathcal{R}} = \mathbb{E}_{q_{\theta_e}(z|x)}\left[-\log p_{\theta_d}\left(x|z\right)\right] = \mathbb{E}_{q_{\theta_e}(z|x)}\left[\frac{1}{2}\left\|\mu_{\theta_d}(z) - x\right\|^2\right]$ can be interpreted as a *reconstruction* error and is estimated by Monte-Carlo method (usually with a single sample), and the term $\mathcal{L}_{\mathcal{P}} = \text{KL}\left(q_{\theta_e}\left(z|x\right) \| \, p(z)\right)$ forces the distribution of the latent space to match the *prior* $p(z)$. Usually, $p(z)$ is a standard Gaussian distribution

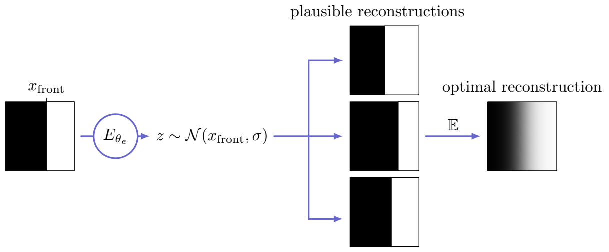
plausible reconstructions
x_{front}
E_{θ_e}
$z \sim \mathcal{N}(x_{\text{front}}, \sigma)$
optimal reconstruction
$\mathbb{E}$

Fig. 1. Illustration on why VAEs produce blurry reconstructions. Consider the example of a binary frontier in an image i and a latent code z which corresponds to the position of the frontier x_{front}. If $q_{\theta_e}(z|i) = \mathcal{N}(x_{\text{front}}, \sigma)$ then $p_{\theta_e}(x_{\text{front}}|z) = \mathcal{N}(z, \sigma)$ and the optimal reconstruction of the pixel at position x is $\mathbb{E}\left[pixel(x)|z\right] = 1 \times \mathbb{P}_{\theta_e}(x > z) + 0 \times \mathbb{P}_{\theta_e}(x < z) = \frac{1}{2}\left(1 + \text{erf}(\frac{x-z}{\sqrt{2}\sigma})\right)$ which is a smooth transition between black and white instead of a sharp transition in the original binary image.

$\mathcal{N}(z; 0, Id))$. The $\mathcal{L}_{\mathcal{P}}$ term acts as an information bottleneck on the latent produced by the encoder. Indeed:

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_{\mathcal{P}}\right] &= \mathbb{E}\left[\text{KL}\left(q_{\theta_e}\left(z|x\right) || p(z)\right)\right] \\
&= \sum_{i=1}^{N} p(x^{(i)}) \int_{\mathcal{Z}} q_{\theta_e}(z|x^{(i)}) \log \frac{q_{\theta_e}(z|x^{(i)})}{p(z)} dz \\
&= \sum_{i=1}^{N} \int_{\mathcal{Z}} p_{\theta_e}(z, x^{(i)}) \log \frac{p_{\theta_e}(z, x^{(i)})}{p(x^{(i)}) p(z)} dz \\
&= \sum_{i=1}^{N} \int_{\mathcal{Z}} p_{\theta_e}(z, x^{(i)}) \log \frac{p_{\theta_e}(z, x^{(i)}) p_{\theta_e}(z)}{p(x^{(i)}) p(z) p_{\theta_e}(z)} dz \qquad (2) \\
&= \sum_{i=1}^{N} \int_{\mathcal{Z}} p_{\theta_e}(z, x^{(i)}) \log \frac{p_{\theta_e}(z, x^{(i)})}{p(x^{(i)}) p_{\theta_e}(z)} dz \\
&\quad + \int_{\mathcal{Z}} p_{\theta_e}(z) \log \frac{p_{\theta_e}(z)}{p(z)} dz \\
&= I_\theta(x; z) + \text{KL}(p_{\theta_e}(z) || p(z))
\end{aligned}$$

with I the mutual information between x and z. This term thus limits the amount of information about the original image that goes through the latent code and pushes the distribution of the latent code produced by the encoder to match the prior latent code distribution.

Limitations of the VAE framework: Understanding and improving VAE are active subjects of research. Some works have focused on reducing the gap in quality which often exists between reconstructions produced by VAEs and images sampled with them [14]. Others have aimed at learning more interpretable latent space structure [15]. While dealing with interesting issues, these papers are not in line with the problem tackled in this paper related to the lack or realism and the blurry aspect of images generated or reconstructed with VAEs.

As we have seen, $\mathcal{L}_{\mathcal{P}}$ acts as an information bottleneck which limits the information about the original image x that passes through the latent code. This creates an uncertainty on the attributes of the original image x when trying to reconstruct it. This uncertainty combined with the use of the mean square error as a reconstruction error causes the generated images $\mu_{\theta_d}(z)$ to be blurry. Indeed under those circumstances, the optimal value for each pixel of the reconstructed image is its expected value given the information available in the latent code [16] (See Figure 1 for an illustration).

The second aspect that prevents VAE and AE in general to produce realistic samples is the use of a pixel-wise reconstruction error combined with the high dimensionality of the natural image manifold. Indeed, it is often assumed that natural images lie on a low dimensional manifold of image space, in particular because of a strong redundancy at a local scale [17]. This point is globally asserted by empirical evidence [18] but can be mitigated with regard to textures. We argue that textures like wood, hair or waves of the ocean are living in a much higher dimensional manifold. This manifold thus cannot be captured in the low dimensional latent space of generative models even in the absence of an explicit information bottleneck. Indeed, it would require a network with a very high capacity to map the low dimensional latent into a high dimensional manifold. One can convince himself of this fact by considering that hair configuration is the product of the configuration of each individual strand of hair. GANs can partially overcome this problem with mode collapse on textures by generating only a subset of this manifold which is enough to fool the discriminator network. However, the use of a powerful pixel-wise reconstruction error in the case of VAE prevents the decoder from using this strategy leading to unrealistic results.

B. Generative Adversarial Networks

GAN globally consists in training two neural networks with adversarial objectives to generate samples indistinguishable from the samples taken from the dataset. The *generator* network parametrized by θ_g is trained to map a random vector to the data space. The *discriminator* or *critic* network parametrized by θ_c is a classifier that is trained to distinguish real samples from generated ones. The key point is that the generator does not have access to real data and can only improve its parameters through its ability to fool the discriminator. The objective of the critic is:

$$\begin{aligned}
\mathcal{O}_C(\theta_c) &= \mathbb{E}_{x \sim p(x)}\left[\log\left(1 - C_{\theta_c}(x)\right)\right] \\
&\quad + \mathbb{E}_{x \sim p_{\theta_g}(x|z)}\left[\log C_{\theta_c}(x)\right]
\end{aligned} \qquad (3)$$

while the generator tries to fool the critic by minimizing:

$$\mathcal{O}_G(\theta_g) = \mathbb{E}_{x \sim p_{\theta_g}(x|z)}\left[\log\left(1 - C_{\theta_c}(x)\right)\right] \qquad (4)$$

Limitations of the GAN framework: GANs have proven to be very successful for generation tasks but suffer from two major limitations in comparison to VAEs: mode collapse and the absence of an encoder network. *Mode collapse* occurs when, at each step, the generator is able to only produce a few different samples. In its extreme case, the generator only produces one type of sample, that is thus easily recognized by the discriminator. In return, the discriminator does not need real data to train and its feedback to the generator through back-propagation does no longer contain useful information. More commonly, the generator produces a limited number of samples and interpolation of them.

Even when a GAN appears to have attained a good solution, *mode collapse* may have occurred slightly and some modes of the data distribution may be missed by the generator. *Mode collapse* also raises the question of the existence of an acceptable pseudo-inverse mapping of the generator defined on the entire dataset space. The second issue is that the GAN framework does not provide an explicit model to find the latent space fibers of samples as it does not have an encoder.

III. Related Works

To leverage both the advantages of GANs and VAEs, [11] proposed the VAE/GAN architecture which combines them. They propose to add a discriminator to push reconstructions from the VAE toward more realism and replaced the standard reconstruction error by a perceptual similarity metric based on the filters learned by the discriminator. This approach is problematic because the discriminator is trained to predict whether an image is a real one or a fake one. Thus, the features extracted from it may not be adapted to describe image content making them a disputable choice to base a similarity metric on. As an example, we noticed that VAE/GAN sometimes fails to reconstruct precisely skin color on the CelebA dataset (see Figure 7) as this information might be useless to some extent for the discriminator. If carefully tuned, this approach tends to work well in practice and allows sharper reconstructions. Nevertheless, [13] pointed out that this approach also tends to exhibit a compromise between VAE and GAN and produces less realistic samples than GAN. They propose the BiGAN architecture [12] which is composed of an encoder that transforms real images into latent codes, a generator that transforms latent codes sampled randomly into images and a discriminator which tries to guess the origin of a couple of image/latent. While this approach is very elegant and produces samples of the same quality as GANs, it is aimed at finding good feature representations in an unsupervised way and often fails to produce very accurate reconstructions. In [19] and [20], the authors propose variations of the BiGAN framework and additional theoretical insights about the latter. They produce more accurate reconstructions in terms of MSE but they are blurry (no hair texture when trained on faces images) which is precisely the issue we aim at solving here. In [2], the authors propose a variation of the VAE/GAN framework where the encoder and the discriminator network are a unique model. While it is not clear why this choice is a good one or not, the model reconstruction loss is the combination between a pixel-wise error and the VAE/GAN reconstruction loss which introduces a compromise between the blurriness of the reconstructions and the features reconstruction fidelity. Similarly, [21] have proposed an elegant framework where the discrimination is made on the latent space. Our approach introduces a reconstruction loss that does not interfere with the realism of the images while being linked with the MSE. By combining our reconstruction loss with adversarial training, we are able to produce photo-realistic reconstructions with no compromise on fidelity. Moreover, our framework is theoretically grounded and is not limited to image data as we show on a toy example (Section V-A) that it can be used in a more general context.

IV. The AVAE framework

A. Complementarity between VAE and GAN

Despite their differences, we show that VAE and GAN exhibit some form of complementarity and that we can build a hybrid approach that solves several problems listed above. One naive hybridization could be to train a VAE with an additional adversarial loss term to push reconstructions toward more realism. However, as we have seen, optimal reconstructions are not always realistic. This approach would lead to choosing a trade-off between reconstruction accuracy and realism as both have conflicting objectives. One of the contributions of this paper is to show that we can derive two complementary losses from the VAE and GAN frameworks which share an optimal solution allowing accurate and realistic reconstructions. In the GAN framework, we can derive a *manifold* loss $\mathcal{L}_{\mathcal{M}}$ from the discriminator network which judges the realism of a given sample. This loss can be interpreted as a "distance" between the data manifold and a sample as described in [22]. In the VAE framework, we train an encoder which maps data in a latent space $\mathcal{Z}$. This latent space can be seen as a map of the data manifold. Distances in the latent space can be interpreted as a distance between two points of the data manifold. This loss is noted $\mathcal{L}_{\mathcal{Z}}$. Our intuition, depicted by Figure 2, is that these two losses can be used in conjunction to train a model which produces realistic images while keeping approximately the latent space organization of a VAE.

We give here further explanation on why the VAE framework fails to produce realistic images and what conditions a reconstruction error should satisfy to achieve accurate and realistic reconstructions. Let us consider an auto encoder that uses a reconstruction error of the form $\mathcal{L}(x,y) = \|x-y\|^2$. Let us note x the input, z the output of the encoder E_{θ_e} and $\hat{x}$ the output of the decoder D_{θ_d}. With the parameters of the encoder fixed, the optimal reconstruction should minimize the expected cost over the potential images $\tilde{x}$ that could have produced the observed z. i.e.

$$\hat{x}^*(z) \in \operatorname*{argmin}_{\hat{x}} \mathbb{E}_{\tilde{x}\sim p_{\theta_e}(\tilde{x}|z)}\left[\|\tilde{x}-\hat{x}\|^2\right] \tag{5}$$

Thus the optimal solution is given by $\hat{x}^*(z) = \mathbb{E}_{\tilde{x}\sim p_{\theta_e}(\tilde{x}|z)}[\tilde{x}]$. The problem is that, in this case the optimal reconstruction $\hat{x}^*$ is the expected value of all the possible reconstructions given the knowledge of the latent code. It leads to a blurry reconstruction, quite unlikely under the data distribution $p_{\mathcal{D}}$ (i.e. $p_{\mathcal{D}}(\hat{x}^*)$ is small).

In a more general setting we can consider objectives of the form: $L(x,a) = ||f(x)-a||^2$ where f is an arbitrary differentiable function and a is a random variable. In this case, the optimal solution verifies:

$$f(\hat{x}^*(z)) = \mathbb{E}_{a\sim p_{\theta_e}(a|z)}[a] \tag{6}$$

This objective has a common optimum with the GAN objective, if and only if we have $p(f(x^*(z))) = p(f(x))$ for $z \sim p(z)$ and $x \sim p_{\mathcal{D}}(x)$. However, to be what we can call a good reconstruction error, $f(x)$ should also carry the maximum of

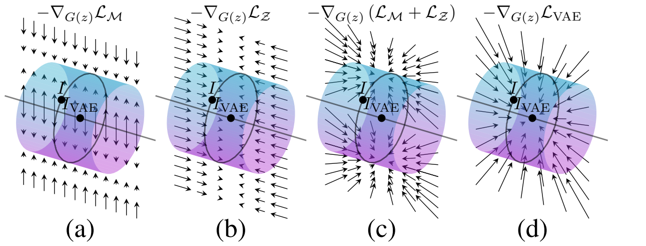
$-\nabla_{G(z)}\mathcal{L}_{\mathcal{M}}$
$-\nabla_{G(z)}\mathcal{L}_{\mathcal{Z}}$
$-\nabla_{G(z)}(\mathcal{L}_{\mathcal{M}}+\mathcal{L}_{\mathcal{Z}})$
$-\nabla_{G(z)}\mathcal{L}_{\text{VAE}}$
I
I_{VAE}
(a)
(b)
(c)
(d)

Fig. 2. The figure depicts a small portion of data space. The cylinders symbolize the real data high-dimensional manifold, the black line the low-dimensional manifold on which the reconstructions of VAEs are restricted. The images on the black circle where the image I is located are all mapped to the same latent code by the encoder network. Thus, they share a common reconstruction: I_{VAE}. This reconstruction is outside of the data manifold as it is the expected value of the original image given the latent code computed by the encoder which is blurry. The arrows represent the gradient of different losses (w.r.t the reconstruction) that are minimized during training: (a) the loss derived from the GAN framework that pushes the reconstructions toward the data manifold, (b) the loss derived from the VAE framework that pushes the reconstructions toward a region where images are mapped to the same latent code by the encoder, (c) their combination and (d) the VAE reconstruction loss i.e. the mean square error. (Note that gradients are represented on a single plane, while there is a radial symmetry around the black line)

information about f. Indeed we could otherwise choose $f = a$ but it would be useless as a reconstruction error.

B. Architecture

Similarly to VAE, the proposed AVAE framework is based on an encoder E_{θ_e} and a decoder D_{θ_d}. We add two additional models: a generator G_{θ_g} and a critic C_{θ_c}. The role of the generator is to produce realistic samples from latent codes.

The VAE part of our framework is similar to classical VAE: it is a parametrized model $q_{\theta_e}(z|x) = \mathcal{N}(z; \mu_{\theta_e}(x), \Sigma_{\theta_e})$ with Σ_{θ_e} a diagonal matrix of the form $\text{diag}(\sigma^2_{\theta_e})$. The prior distribution of the latent codes is $p(z) = \mathcal{N}(z; 0, Id)$ and $p_{\theta_d}(x|z) = \mathcal{N}(x; \mu_{\theta_d}(z), Id)$. With such choices, $\mathcal{O}_{\text{VAE}}(\theta_e, \theta_d; x)$ can be estimated by a Monte-Carlo method. Indeed, the Kullback-Leibler divergence term of the loss $\text{KL}(q_{\theta_e}(z|x) \| p(z))$ is equal to:

$$\frac{1}{2} \sum_{j=1}^{\dim(\mathcal{Z})} \sigma^2_{\theta_e j} + \mu^2_{\theta_e}(x)_j - 1 - \log \sigma^2_{\theta_e j} \tag{7}$$

The reconstruction term of the loss $\mathbb{E}_{q_{\theta_e}(z|x)}[\log p_{\theta_d}(x|z)]$ can be estimated by Monte-Carlo, sampling z from $q_{\theta_e}(z|x)$ and noting that:

$$\log p_{\theta_d}(x|z) = -\frac{\dim(x)}{2} \log 2\pi - \frac{1}{2} \left\| \mu_{\theta_d}(z) - x \right\|^2 \tag{8}$$

z being sampled from $q_{\theta_e}(z|x)$, the loss of the VAE for one sample is the following (without constant terms):

$$\begin{aligned} \mathcal{L}_{\text{VAE}}(\theta_e, \theta_d; x) &= \frac{1}{2} \left\| \mu_{\theta_d}(z) - x \right\|^2 \\ &+ \frac{1}{2} \sum_{j=1}^{\dim(\mathcal{Z})} \sigma^2_{\theta_e j} + \mu^2_{\theta_e}(x)_j - \log \sigma^2_{\theta_e j} \end{aligned} \tag{9}$$

For the generator part, when we want to use it for reconstruction, we build its input by concatenating z the latent code produced by the encoder with a random vector ξ sampled from $\mathcal{N}(0, Id)$ to form the latent code for our generator. z encodes the information captured by the encoder while ξ encode the variation not captured by it. With this choices, we sample from $p_{\theta_g}(x|z)$ by taking $x = G_{\theta_g}(z, \xi)$. Note that ξ can be removed if we consider that for a given z there is only one possible reconstruction but we present here the general setting as we consider. To sample a random image from the generator we simply sample z from the prior distribution defined in the VAE part and ξ from $\mathcal{N}(0, Id)$. Ideally, the generator should invert the encoder and thus $p_{\theta_g}(x|z)$ should be as close as possible than $p_{\theta_e}(x|z)$. This consideration leads us to minimizing the following negative log likelihood with $z \sim \mathcal{N}(0, Id)$ and $x \sim p_{\theta_g}(x|z)$:

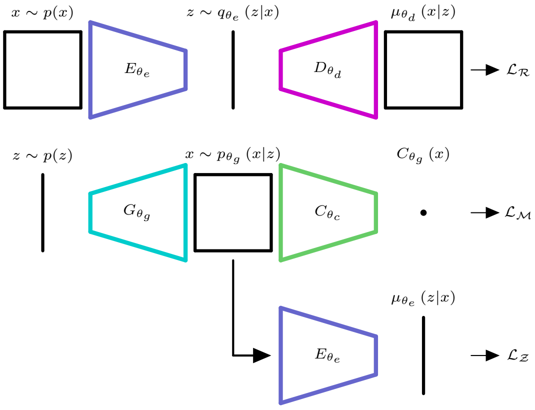

Fig. 3. Summary of our Adversarial Variational Auto Encoder framework. $E_{\theta_e}, D_{\theta_d}, G_{\theta_g}, C_{\theta_c}$ are respectively the encoder, decoder, generator and critic (discriminator). Note that the weights of the encoder E_{θ_e} are shared between the two architectures. Images are denoted by the letter x and latent codes by the letter z.

$$\begin{aligned} \mathcal{L}_G(\theta_g) &= \mathbb{E}\left[-\log p_{\theta_e}(x|z)\right] \\ &= \mathbb{E}\left[-\log p_{\theta_e}(z|x) p(x)\right] + C \\ &= \mathbb{E}\left[-\log p_{\theta_e}(z|x)\right] + \mathbb{E}\left[\log \frac{p_{\theta_g}(x)}{p(x) p_{\theta_g}(x)}\right] + C \\ &= \mathbb{E}\left[-\log p_{\theta_e}(z|x)\right] + \text{KL}(p_{\theta_g}(x) \| p(x)) + H_{\theta_g} + C \end{aligned} \tag{10}$$

with H_{θ_g} the differential entropy of the distribution $p_{\theta_g}(x)$. The term $\log p_{\theta_e}(z|x)$ can be computed directly:

$$\begin{aligned} \log p_{\theta_e}(z|x) &= \log \mathcal{N}(z; \mu_{\theta_e}(x), \Sigma_{\theta_e}) \\ &= -\frac{\dim(\mathcal{Z})}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_{\theta_e}| \\ &\quad - \frac{1}{2} \left\| \frac{\mu_{\theta_e}(x) - z}{\sigma_{\theta_e}} \right\|^2 \end{aligned} \tag{11}$$

We define the reconstruction loss $\mathcal{L}_{\mathcal{Z}}$ by removing constant terms in Equation 11:

$$\mathcal{L}^a_{\mathcal{Z}}(\theta_g; z, \theta_e) = \frac{1}{2} \left\| \frac{\mu_{\theta_e}(x) - z}{\sigma_{\theta_e}} \right\|^2 \tag{12}$$

We can estimate the second term by training a classifier C that discriminates generated images from real ones by minimizing the cross-entropy:

$$\begin{aligned} \mathcal{L}_C(\theta_c) = &- \mathbb{E}_{x \sim p(x)}\left[\log\left(1 - C_{\theta_c}(x)\right)\right] \\ &- \mathbb{E}_{x \sim p_{\theta_g}(x|z)}\left[\log C_{\theta_c}(x)\right] \end{aligned} \tag{13}$$

```
Initialize parameters of the models: $\theta_e$, $\theta_d$, $\theta_g$, $\theta_c$
while training do
    {Forward pass.}
    $x^{\text{real}} \leftarrow$ batch of images sampled from the dataset.
    $z_\mu^{\text{real}}, z_\sigma^{\text{real}} \leftarrow E_{\theta_e}(x^{real})$
    $z^{\text{real}} \leftarrow z_\mu^{\text{real}} + \epsilon z_\sigma^{\text{real}}$ with $\epsilon \sim \mathcal{N}(0, Id)$
    $\mu^{\text{real}} \leftarrow D_{\theta_d}(z^{\text{real}})$
    $x^{\text{fake}} \leftarrow G_{\theta_g}(z^{\text{fake}}, \xi)$ with $z^{\text{fake}}, \xi \sim \mathcal{N}(0, Id)$
    $z_\mu^{\text{fake}}, z_\sigma^{\text{fake}} \leftarrow E_{\theta_e}(x^{\text{fake}})$
    $C^{\text{real}}, C^{\text{fake}} \leftarrow C_{\theta_c}(x^{\text{real}}), C_{\theta_c}(x^{\text{fake}})$
    {Compute losses gradients and update parameters.}
    $\theta_e \overset{-}{\leftarrow} \nabla_{\theta_e}\mathcal{L}_{\text{VAE}}(\theta_e, \theta_d)$ ; $\theta_g \overset{-}{\leftarrow} \nabla_{\theta_g}\mathcal{L}_G(\theta_g)$
    $\theta_d \overset{-}{\leftarrow} \nabla_{\theta_d}\mathcal{L}_{\text{VAE}}(\theta_e, \theta_d)$ ; $\theta_c \overset{-}{\leftarrow} \nabla_{\theta_c}\mathcal{L}_C(\theta_c)$
end while
```

Fig. 4. Algorithm to train the Adversarial Variational Auto Encoder.

GENERATOR

Dense
units: $w * 128$,
Reshape
new size: $(4, 4, w * 8)$
Batch normalization
ReLU
Up-block
channels: $w * 4$
Up-block
channels: $w * 2$
Up-block
channels: w
Transposed convolution
channels: 3, stride: 2
Tanh

DOWN-BLOCK

Convolution
stride: 2, no bias
Batch normalization
ReLU

CRITIC

Convolution
channels: w,
stride: 2
Leaky-ReLU
Down-block
channels: $w * 2$
Down-block
channels: $w * 4$
Down-block
channels: $w * 8$
Dense
units: 1
Sigmoid

Fig. 5. Generator and critic architectures. The decoder architecture is identical to the generator architecture and the encoder architecture differs from the critic architecture by the number of units in the last layer and by the absence of a Sigmoid activation at the end. Up-block is similar to Down-block but with transposed convolutions instead of convolutions and ReLUs instead of leaky-ReLUs. All convolutions and transposed convolutions share the same filter size (5) and use 'same' padding. σ_z is chosen independent of x and is learned directly. w is a width multiplier (we typically use $w = 128$). For the BiGAN implementation, we use a two-hidden-layer MLP for the latent code inputs and a critic-style architecture for the image inputs. The two outputs representations are then concatenated and used as input of a two-hidden-layers MLP.

Under this loss, the optimal solution for C is:

$$C^* : x \rightarrow \frac{p_{\theta_g}(x)}{p(x) + p_{\theta_g}(x)} \tag{14}$$

$\mathcal{L}_\mathcal{M}$ is then defined by sampling x from $p_{\theta_g}(x)$. Hence:

$$\mathcal{L}_\mathcal{M}(\theta_g; x, \theta_e) = \operatorname{logit} C \tag{15}$$

Indeed, $\operatorname{logit} C \approx \operatorname{logit} C^* = \log\left(\frac{p_{\theta_g}(x)}{p(x)}\right)$ which is an unbiased estimator of the Kullback-Leibler divergence term. Minimizing the differential entropy H_{θ_g} of the distribution $p_{\theta_g}(x)$ will push it to be as peaked as possible and is not data dependent. Moreover, this term is intractable. Hence, as a form of regularization, we remove it. One problem still remains. Indeed the optimal reconstruction for $\mathcal{L}_\mathcal{Z}^a$ verifies the following equation: $\mu_{\theta_e}(\hat{x}^*(z)) = z$ and thus $p(\mu_{\theta_e}(\hat{x}^*(z))) = \mathcal{N}(\mu_{\theta_e}(\hat{x}^*(z)); 0, I)$ while $p(\mu_{\theta_e}(x)) = \mathcal{N}(\mu_{\theta_e}(x); 0, I - \Sigma)$. To solve this problem, we propose to replace the expression of $\mathcal{L}_\mathcal{Z}^a$ by:

$$\mathcal{L}_\mathcal{Z}^b(\theta_g; z, \theta_e) = \frac{1}{2}\left\|\frac{\mu_{\theta_e}(x) - \sqrt{1 - \sigma_{\theta_e}^2}\, z}{\sigma_{\theta_e}}\right\|^2 \tag{16}$$

With this loss the optimal solution $\hat{x}^*(z)$ verifies $\mu_{\theta_e}(\hat{x}^*(z)) = \sqrt{1 - \sigma_{\theta_e}^2}\, z$ thus $p(\mu_{\theta_e}(\hat{x}^*(z))) = \mathcal{N}(\mu_{\theta_e}(x); 0, I - \Sigma) = p(\mu_{\theta_e}(x))$ as we have seen, its ensure that this loss has a common optimum with the GAN objective. This new loss takes into account the fact that when σ_{θ_e} is large, the observed z is mostly noise and $\mu_{\theta_e}(x)$ is close to zero. The loss resulting from these considerations is $\mathcal{L}_G = \mathcal{L}_\mathcal{Z}^b + \mathcal{L}_\mathcal{M}$. It combines a GAN type loss $\mathcal{L}_\mathcal{M}$ and a reconstruction loss on the latent codes $\mathcal{L}_\mathcal{Z}$ which is similar to that described in Section IV-A. The AVAE framework is globally presented in Figure 3, with the relations between its components, and Figure IV-B gives the algorithm to train it. From a GAN perspective, the method can be viewed as constraining the latent space organization of the generator with the encoder model. It thus limits to some point the problem of mode collapse as the reconstruction error on the latent code prevents the generator to produce similar samples. As a consequence, it counteracts the mechanism pointed out by [10] to explain mode collapse by pushing generated samples apart from each other. The proposed architecture differs from VAE/GAN on several important aspects. The decoder and generator are separated in our work and our reconstruction error is based on the encoder model and not on the discriminator as in VAE/GAN to ensure that the error is informative about the image content.

V. Experimental results

Datasets: We evaluate the models on six image datasets: LSUN bedroom [23] (64x64 images of bedrooms), CelebA [24] (64x64 faces cropped images), FFHQ dataset (256x256 faces) [9], CIFAR10, CIFAR100 [25] (32x32 images of 10 and 100 categories) and SVHN [26] (32x32 images of house numbers images). Images are resized to the sizes mentioned above and CelebA images are center-cropped at 70%.

Implementation details: all the low resolution experiments have been conducted with Tensorflow 2.0 [27] on an NVIDIA GTX 1080 Ti GPU with 11Go of memory. Full code will be available on github. All models share similar architecture blocks, inspired by [28], to allow a fair comparison. Architecture details are presented in Figure 5. Each model is trained with hyper-parameters recommended in [28] for $5e4$ iterations with a batch size of 64. Because the reconstruction loss of the VAE part of VAE/GAN is a perceptual loss which differs from the MSE used in our model and in the classical VAE, the balance between the Kullback-Leibler divergence term and the reconstruction term in the VAE loss is not the same between models. We observed that the Kullback-Leibler divergence term is usually much higher for the VAE/GAN model which indicates

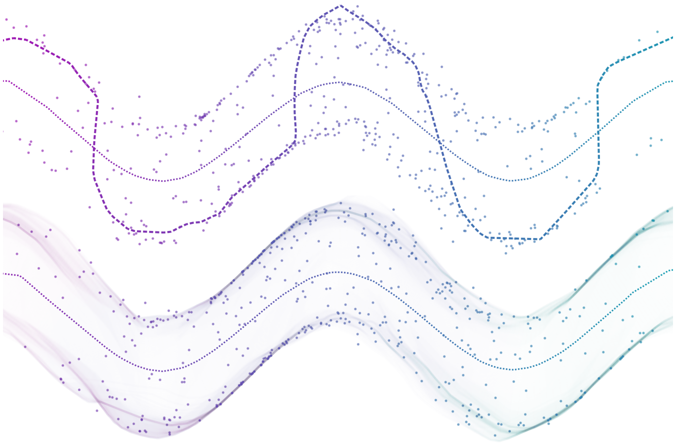

Fig. 6. Illustration of a toy example with two-dimensional data and a one-dimensional latent space. Points: data, dotted line: manifold of reconstructions from VAE, dashed line/density: manifold of reconstruction with our model. Color encodes the position in the one-dimensional latent space. Top: with a deterministic generator. Bottom: with a probabilistic generator. (best seen with zoom and color)

that it conveys much more information in its latent code and thus introduces a bias in the reconstruction performance comparison between models. To solve this problem, we introduced a hyper-parameter β to weight the Kullback-Leibler divergence in the encoder loss as in [29] in order to get similar Kullback-Leibler divergences. This hyper-parameter search leads us to the following ($\beta_{\text{LSUN bedroom}} = 4$, $\beta_{\text{celeba}} = 5$, $\beta_{\text{CIFAR10}} = 10$, $\beta_{\text{CIFAR100}} = 10$, $\beta_{\text{SVHN}} = 20$). The high resolution experiment was conducted with a network architecture derived from the StyleGAN V2 architecture [30] trained on 8 NVIDIA Quadro P5000 GPUs.

A. *Toy dataset*

We begin by testing our approach on a toy dataset to validate the theory. The dataset is composed of 2D points generated from two generative factors z_1 and z_2. The data generation procedure is the following: $z_1, z_2, \epsilon \sim \mathcal{N}(0, 1)$ and $x = f(z_1, z_2, \epsilon) = (3z_1 + 0.1\epsilon, cos(3z_1 + \tanh(3z_2)) + 0.1\epsilon)$. For the model, we use a latent space of dimension one to simulate the problem of the low dimensionality of the latent space compared to the high dimensionality of the data manifold. Models are two-hidden-layer perceptrons with 128 units. Models are trained with the method described proposed in this paper. We then draw the manifold of the generated points to see how the model behave compared to a VAE. Results of this experiment can be seen in Figure 6 where we can see that reconstructions from the VAE are in a region of low likelihood of the data distribution while AVAE reconstructions follow the shape of the VAE manifold while covering regions of higher likelihood. It shows that our model is able to produce realistic reconstructions even when the latent code do not contain all the information needed to reconstruct the original image perfectly. Here there is an ambiguity as we do not know if the original sample is from the top distribution or the bottom one given a latent code corresponds to two. In order to produce a realistic result the generator has to make an arbitrary choice. Our approach allows the generator to make such choice while the decoder from the VAE outputs the average of possible choices resulting in an unlikely/unrealistic reconstruction. On the same Figure we can see that when using a stochastic generator with additional latent variables, it learns to generate missing regions of the data distribution while keeping the VAE latent space structure.

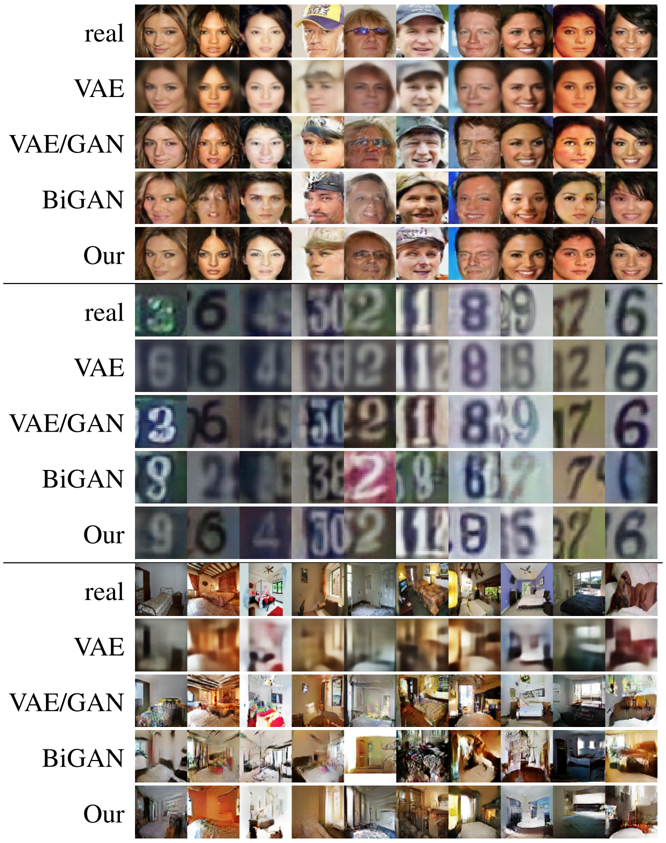

Fig. 7. Qualitative comparison of the quality of reconstructions between several frameworks namely VAE, VAE/GAN, BiGAN and our model on three datasets: CelebA, SVHN and LSUN bedroom.

B. *Qualitative results*

Here, we present some qualitative results on the CelebA SVHN and LSUN bedroom datasets. A comparison of samples reconstruction between our model and other models is presented in Figure 7. We also present a visual comparison of samples generated by our model and other generative models in Figure 8. Additional qualitative results will be available on github. We can see on these figures that generated images are of comparable quality of GAN generated images for both generation and reconstructions. VAE reconstructions and generated samples look blurry, BiGAN generated images are of good quality but reconstructions are not accurate. VAE/GAN produces both good reconstructions and generated samples. However, while our judgment is subjective, we find that reconstructions produced by VAE/GAN are less accurate than ours and images are less realistic than with GAN, BiGAN or our approach.

One may notice that for the LSUN bedroom dataset, reconstructions produced by our model are not convincing. However, we can explain this by the very poor performance of the VAE suggesting that not enough information passes through the latent code to create a reconstruction visually close to the

TABLE I

Reconstruction errors (MSE and LPIPS [31]) and FID [32] of generated images for different models. Lower values are better for all metrics. Reported results are the average and standard deviation over five runs.

		Bedroom	CelebA	CIFAR10	CIFAR100	SVHN
VAE	MSE	0.06 ± 0.00	0.03 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.02 ± 0.00
	LPIPS	0.58 ± 0.00	0.18 ± 0.00	0.26 ± 0.00	0.25 ± 0.00	0.08 ± 0.00
	FID	229.75 ± 1.45	60.04 ± 0.47	136.75 ± 0.57	129.71 ± 1.01	68.16 ± 2.10
GAN	FID	110.59 ± 19.55	14.54 ± 0.41	32.01 ± 0.41	34.51 ± 0.59	23.83 ± 3.99
VAE/GAN	MSE	0.18 ± 0.01	0.07 ± 0.00	0.14 ± 0.02	0.15 ± 0.02	0.06 ± 0.02
	LPIPS	0.26 ± 0.01	0.09 ± 0.00	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.02
	FID	60.02 ± 2.36	26.45 ± 4.66	39.04 ± 2.42	40.03 ± 0.71	17.02 ± 2.58
BiGAN	MSE	0.42 ± 0.05	0.18 ± 0.01	0.31 ± 0.02	0.33 ± 0.01	0.12 ± 0.01
	LPIPS	0.44 ± 0.02	0.16 ± 0.00	0.14 ± 0.00	0.16 ± 0.00	0.12 ± 0.01
	FID	91.72 ± 18.10	18.49 ± 5.06	34.61 ± 1.29	35.40 ± 1.23	27.77 ± 2.96
Ours with ξ with $\mathcal{L}^a_{\mathcal{Z}}$	MSE	0.12 ± 0.00	0.05 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.04 ± 0.00
	LPIPS	0.36 ± 0.00	0.11 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.10 ± 0.00
	FID	85.11 ± 2.87	16.99 ± 0.58	33.65 ± 0.28	39.81 ± 0.60	27.64 ± 2.41
Ours without ξ with $\mathcal{L}^a_{\mathcal{Z}}$	MSE	0.12 ± 0.00	0.05 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.04 ± 0.00
	LPIPS	0.35 ± 0.00	0.11 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.09 ± 0.00
	FID	84.29 ± 5.28	16.23 ± 0.50	33.49 ± 0.50	38.69 ± 0.62	28.47 ± 8.24
Ours without ξ with $\mathcal{L}^b_{\mathcal{Z}}$	MSE	0.12 ± 0.00	0.05 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.04 ± 0.00
	LPIPS	0.35 ± 0.00	0.11 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.08 ± 0.00
	FID	80.99 ± 1.82	15.01 ± 0.82	33.67 ± 0.61	38.35 ± 0.57	21.11 ± 0.42

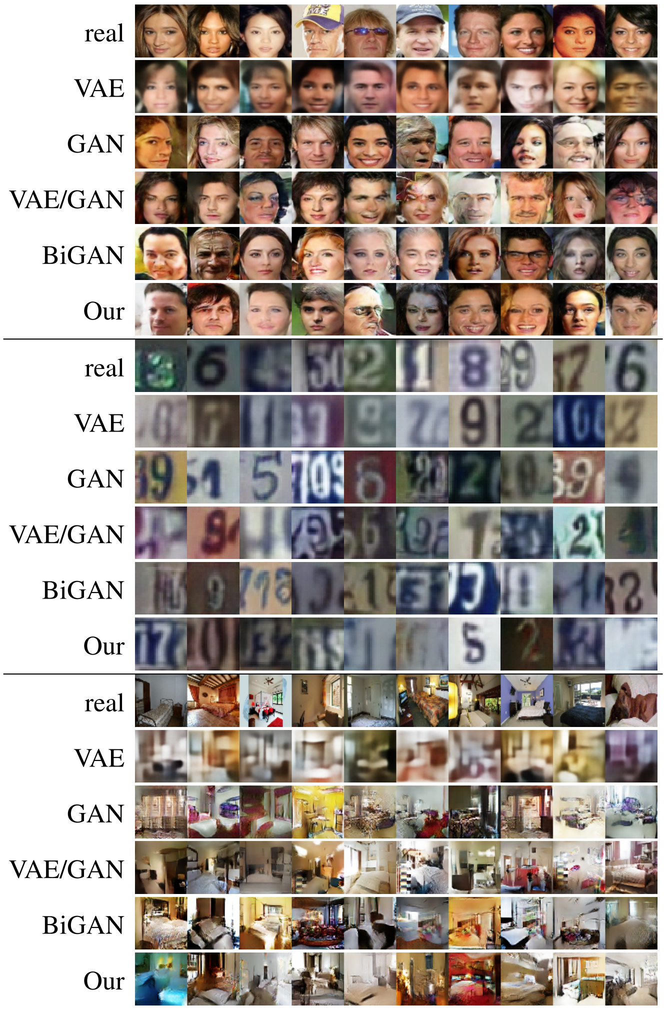

Fig. 8. Generated images for randomly sampled latent codes for CelebA, SVHN and LSUN bedroom.

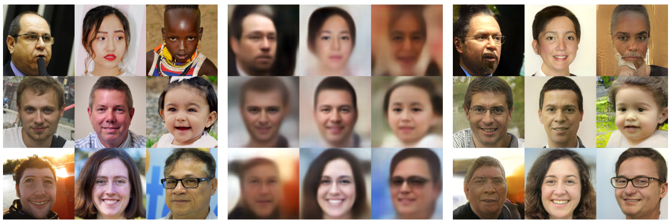

Fig. 9. Qualitative results on high resolution images. left to right: original images, reconstructions with the VAE decoder, reconstructions with the generator. This figure shows that with a limited amount of information the decoder fails to produce realistic reconstruction while our generator is capable of it. Note that the fidelity of the reconstruction is ultimately limited by the information contained in the latent code produced by the encoder.

original image. However, even here, our model still produces sharp images close to the target ones in terms of MSE showing that our model follows the latent structure of the VAE trained with the MSE as a reconstruction error.

We also conducted an experiment on higher resolution images (256x256 FFHQ face images) to see if our method can be scaled to high resolution images. To conduct this experiment, we made straightforward modifications to the style GAN V2 [30] using the approach proposed here. Results of this experiments are presented in Figure 9. These results confirm the scalability of the proposed approach to bigger architectures.

C. *Quantitative results*

To quantitatively evaluate the performance of our method, we selected several metrics. The quality of the reconstructed images is evaluated by the Mean Squared Error or MSE and the LPIPS [31]. We use the FID [32] to measure the realism of generated images. A comparison between VAE, GAN, BiGAN [13], [12], VAE/GAN [11], and our model is presented in Table I. Reconstructions errors are computed on validation images not used during training, namely the *test* or *validation* splits of TensorFlow datasets. FID is computed over 50000 randomly generated samples and compared to training data samples as FID requires a lot of samples to be calibrated. It must be noted that some metrics are biased toward some architectures: the MSE is favorable to the VAE model because it is the loss used to train it. It is also the case for our approach, as information contained in the latent code is optimized to produce accurate reconstructions in terms of MSE. VAE/GAN is also advantaged in terms of LPIPS and FID as this model uses a perceptual similarity metric based on a classifier as a reconstruction error and the FID and LPIPS are also based on deep features. Globally, our model exhibits a good compromise between accurate reconstructions (MSE and LPIPS) and realism (FID), thus combining the best of VAE and GAN.

VI. Discussion

The proposed framework can be used to generate images from a pre-trained representation. Thus, it is not a feature learning method and only features learned by the VAE are described by the representation. However, while we focused on a VAE architecture to produce the latent representation, our approach can be further extended. Indeed one could for example train a classifier while constraining its last feature layer in the same way the latent code is constrained and use it as a latent code in our method in order to focus on different features of the image. One could even concatenate several of these representation to train a model which fits their needs. We keep this extension as a potential future work.

Acknowledgment This paper was supported by European Union´s Horizon 2020 research and innovation programme under grant number 951911 - AI4Media. This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” *arXiv e-prints*, p. arXiv:1406.2661, Jun 2014.

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Neural Photo Editing with Introspective Adversarial Networks,” in *International Conference on Learning Representations*, Apr 2017.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” *arXiv e-prints*, p. arXiv:1703.10593, Mar 2017.

[4] R. Huang, S. Zhang, T. Li, and R. He, “Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis,” *arXiv e-prints*, p. arXiv:1704.04086, Apr 2017.

[5] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks,” *arXiv:1612.03242*, Dec 2016.

[6] A. Plumerault, H. Le Borgne, and C. Hudelot, “Controlling generative models with continuous factors of variations,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1laeJrKDB

[7] D. P. Kingma and M. Welling, “Auto-encoding variational bayes.” in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13

[8] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” *arXiv e-prints*, p. arXiv:1809.11096, Sep 2018.

[9] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *arXiv e-prints*, p. arXiv:1812.04948, Dec 2018.

[10] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks,” *arXiv preprint arXiv:1611.02163*, 2016.

[11] A. Boesen Lindbo Larsen, S. Kaae Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” *arXiv e-prints*, p. arXiv:1512.09300, Dec 2015.

[12] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial Feature Learning,” *arXiv e-prints*, p. arXiv:1605.09782, May 2016.

[13] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, “Adversarially Learned Inference,” *arXiv e-prints*, p. arXiv:1606.00704, Jun 2016.

[14] B. Dai and D. Wipf, “Diagnosing and enhancing VAE models,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1e0X3C9tQ

[15] A. Klushyn, N. Chen, R. Kurle, B. Cseke, and P. van der Smagt, “Learning hierarchical priors in vaes,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 2870–2879.

[16] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” *arXiv:1511.05440*, Nov 2015.

[17] E. R. Kretzmer, “Statistics of television signals,” *Bell System Technical Journal*, vol. 31, no. 4, pp. 751–763, 1952.

[18] D. L. Ruderman, “The statistics of natural images,” *Network: Computation in Neural Systems*, vol. 5, no. 4, pp. 517–548, 1994.

[19] L. Chen, S. Dai, Y. Pu, C. Li, Q. Su, and L. Carin, “Symmetric Variational Autoencoder and Connections to Adversarial Learning,” *arXiv e-prints*, p. arXiv:1709.01846, Sep 2017.

[20] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, “ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching,” *arXiv e-prints*, p. arXiv:1709.01215, Sep 2017.

[21] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan, “IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis,” *arXiv e-prints*, p. arXiv:1807.06358, Jul. 2018.

[22] J. Zhao, M. Mathieu, and Y. LeCun, “Energy-based generative adversarial networks,” in *ICLR 2017*, 2017.

[23] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop.” *CoRR*, vol. abs/1506.03365, 2015. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1506.html#YuZSSX15

[24] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *ICCV*, December 2015.

[25] A. Krizhevsky, “Learning multiple layers of features from tiny images,” University of Toronto, Tech. Rep., 2009.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, “Reading digits in natural images with unsupervised feature learning,” *NIPS*, 2011.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org/

[28] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *arXiv e-prints*, p. arXiv:1511.06434, Nov 2015.

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

[30] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” *arXiv e-prints*, p. arXiv:1912.04958, Dec. 2019.

[31] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” *arXiv e-prints*, p. arXiv:1801.03924, Jan 2018.

[32] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” *arXiv e-prints*, p. arXiv:1706.08500, Jun 2017.