Deep Generative Models: VAE and GAN

Mohit Bajaj (95339164)
Department of Computer Science
University of British Columbia
mbajaj01@cs.ubc.ca

Gursimran Singh (98462161)
Department of Computer Science
University of British Columbia
msimar@cs.ubc.ca

Abstract

Deep generative models are at the forefront of the unsupervised learning revolution. The two major approaches are variational autoencoders and generative adversarial networks. In this report, we discuss these models and their interpretations in detail. We highlight the solutions to the major challenges faced in their research. Additionally, we discuss recent research which tries to unify these models and suggest possible future research directions.

1 Introduction

Machine learning algorithms can be classified into two broad categories. Discriminative models learn the conditional probability distribution of labels given the data $p(y|x)$. These models are easy to train, however, they lack the ability to understand the underlying distribution of the data x. Generative models, in contrast, learn the joint probability density p(x,y). The ability to jointly model data x and labels y, allows them to generalize better in case of limited and/ or missing data. They can also be used to compute conditional density using the Bayes rule $p(y|x)$. [26] provides a nice analysis of discriminative vs generative models and the situations in which we prefer one over the other. Additionally, they can also be used to generate new samples (x, y) from the distribution. For instance, generative models are used to generate new hypothetical faces of celebrity, generate more text in a particular handwriting style and new artificial environments to be used in reinforcement learning. Other applications include super-resolution [22], conversion of sketches to images [7], etc.

Typically, we train generative models by maximizing the likelihood of the training data $\theta^* = arg\max_\theta \sum_{i=1}^{N} log(p_{model}(x_i;\theta))$. However, optimizing the maximum likelihood objective often becomes intractable in the general formulation of p_{model}. A wide range of models like Gaussian mixture models, Markov chains, and hidden Markov models makes simplifying conditional independence assumptions to make the objective tractable. However, due to the simplifying assumptions, these models are limited in terms of expressiveness. Hence they are not able to model many data distributions of interest which are often non-linear and highly complex. Deep generative models relax some of these strong assumptions and model the joint density using neural networks as a powerful function approximator. Deep generative models can be further classified as explicit density models and implicit density models based on how we define likelihood, its gradients, and any approximations.

Explicit density models define an explicit function $p_{model}(x;\theta)$ to represent the probability density of the data. Explicit models that use approximations for the intractable likelihood allow us to use a large class of powerful models. Instead of optimizing the intractable likelihood directly, these models optimize a lower bound approximation of the objective. VAEs fall in this category. The main disadvantage of these methods is that when we have a weak approximation of the posterior or the prior distribution, optimizing the lower bound does not effectively learn the target distribution. Implicit density models do not model the probability density explicitly. Instead, it just provides a stochastic procedure to generate data. GANs fall under this category. These models are trained by directly sampling from the p_{model} using a game theoretic approach between two networks, the

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

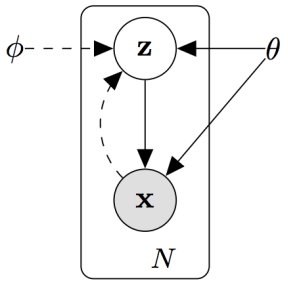

Figure 1: DAG representation of a variational autoencoder (VAE) model. Generative process is denoted by the solid arrow ($p_\theta(x|z)p(z)$) and the variational approximation $q_\phi(z|x)$ is denoted by the dashed lines. The variational parameters ϕ and the generative model's parameters θ are leaned jointly. [20]

generator and the discriminator. However, finding an equilibrium is hard and it makes them hard to train.

In the literature of deep generative models, the two popular approaches are GANs and VAEs. In this report, we provide a perspective of the current state of the research in these models. In Section 2, we introduce VAEs and discuss its use in representation learning, along with challenges and solutions. In Section 3, we discuss GANs, the various challenges we face while training these networks and its applications. In Section 4, we discuss the new frontiers in the field of deep generative models, trying to combine both the VAEs and GANs. Finally, we conclude our discussion in Section 4.2.

2 Variational Autoencoder (VAE)

The variational autoencoder (VAE) [20] is represented by the joint probability density $p(x, z) = p(x|z)p(z)$ as shown in Fig 1. Computing the posterior $p(z|x)$ in this model requires exponential time due to the intractable integral in the calculation of marginal likelihood $p(x) = \int_z p(x|z)p(z)dz$. Variational inference approximates the posterior with a family of distributions $q_\lambda(z|x)$, parametrized by λ. We learn the optimal parameter by minimizing the KL divergence between the two distributions $q^*_\lambda(z|x) = arg\min KL(q_\lambda(z|x)||p(z|x))$. However, even with variational inference, learning the optimal parameter is computationally intractable as it still involves the pesky evidence $p(x)$. The VAE paper [20] suggests maximizing a tractable approximation to the objective known as the Evidence Lower Bound (ELBO).

$$ELBO(\lambda) = \mathbb{E}_q[\log p(x, z)] - \mathbb{E}_q[\log q_\lambda(z, x)] \tag{1}$$

The reason behind optimizing the alternate objective can be seen by rewriting the KL divergence as $ELBO(\lambda) = logp(x) - KL(q_\lambda(z|x)||p(z|x)) >= 0$. Due to Jenson's inequality, the KL term is always greater than or equal to zero, due to which the ELBO can be seen as a lower bound. Hence, it is guaranteed to obtain a value of the original objective which is as high as the maximum of $ELBO(\lambda)$.

In order to minimize this objective using a stochastic gradient estimator, [20] provides two key insights. The first is that we can optimize ELBO objective for each data point separately. This owes to the fact that individual data points do not share latent variables and are independent and identically distributed. Using this idea, we can rewrite the objective Eq.1 in terms of individual data points $ELBO_i(\lambda) = \mathbb{E}_{q\lambda(z|x_i)}[\log p(x_i, z)] - \mathbb{E}_q[\log q_\lambda(z, x_i)]$. The second insight is the reparametrization trick which helps to separate out the stochasticity and write the objective function deterministically in terms of the parameters. In case of normal distribution, instead of sampling from the $N(\mu, \Sigma)$, we can sample from the $N(0, 1)$ and then apply transformations to convert it into a sample with the desired mean and variance. This allows us to backpropagate even through the stochastic network and perform inference.

In the vocabulary of neural networks, we can think of a VAE as a combination of an encoder, a decoder, and a loss function. The encoder $q_\theta(z|x)$ is a neural network which models the posterior probability density of latent variables $p(z|x)$. More concretely, it takes a feature vector x as an input and outputs the hidden representation z. Typically, we allow the hidden layer act as a bottleneck, $z << x$, which helps learn the underlying lower-dimensional latent factors. Although this setup looks like a classical autoencoder, it is quite different in terms of the underlying mathematics. The main difference is that the lower dimensional space is stochastic. So instead of directly outputting the hidden representation, the encoder outputs the parameters of the posterior distribution from which we can sample the hidden representations z. The decoder network $p_\Phi(x|z)$ is again a neural network which models the sampling distribution of data, given the latent variables $p(x|z)$. This network directly outputs the distribution of x, from which we can sample. For instance, in case of handwritten digits, it outputs a Bernoulli parameter for each pixel of the image. We can later use this distribution to sample directly.

2.1 Representation learning using VAE

In contrast to other generative models, VAE's objective is designed not only to sample but also to reproduce a specific data point. Since each data point is passed through a bottleneck layer z, we can think it as learning meaningful concepts in the data: a lower dimensional coding distribution. Hence, an important application of VAE is representation learning. However, in traditional VAEs, there are three main challenges associated with representation learning. We will discuss these in the following sections along with the potential solutions.

2.1.1 Entanglement of codes

The variational lower bound objective of the VAE does not penalize dependence between the dimensions of the coding distribution. This leads to entangled codes, where every dimension of the coding distribution depends on one or more generative factors. However, it has been suggested that learning disentangled codes is useful for a variety of tasks and domains [3, 29]. [16] introduces a state-of-the-art framework β-VAE, which modifies the vanilla VAE objective by introducing a new hyperparameter β as $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x))||p(z)$. The new objective allows us to learn statistically independent latent factors by increasing the weight on the regularization term. The intuition behind this lies in the interpretation of the KL divergence term in the VAE objective. The KL term penalizes the network for learning coding distributions which look significantly different from the isotropic Gaussian. [6] presents the key intuitions behind the effects on the coding distribution, as we increase the weight β on the KL term. The first effect is **data-locality**, which incentivizes the reconstructions to change as smoothly as possible with a small change in the code. The second is **dimentional efficiency**, which incentivizes the coding distribution to encode the most salient factors in as few dimensions as possible. The third is **axis-alignment**, which incentivizes the coding distribution to remain independent and allow major generative factors to be aligned with the dimensions of z.

2.1.2 Ignored codes

The KL term in the ELBO-VAE objective encourages the amortized inference distribution $q(z|x)$, for each data point x, to be close to an isotropic Gaussian. While this is essential for a variety of reasons discussed above, another line of research finds this too restrictive [9, 32, 5]. With a sufficiently expressive conditional distribution, like a pixelCNN autoregressive density estimator [33, 12], the ELBO-VAE objective finds an undesirable solution where the latent code gets completely ignored. The conditional distribution can perfectly model the p_{data}, with the same distribution $p_\theta(x|z); z \in \mathbb{Z}$ and hence the amortized inference distribution can default to an uninformative isotropic Gaussian. This leads to vanishing small mutual information between x and z, a problem is known as **information preference problem** [9]. A potential solution to the problem has been addressed in [14], which aims to restrict the conditional distribution but the approach is limited and causes additional overhead. Even with a less complex conditional distribution, the restrictive KL-term in the ELBO-VAE objective tends to ignore the latent codes. [35] proposes a family of VAEs with a new objective to maximize the mutual information between x and z. We will discuss this in the sections below.

2.1.3 Exploding latent space

Another problem introduced in [35] is that the ELBO-VAE tends to over-fit the data and leads to infinite variance in the $q(z)$ distribution. As a result, the amortized inference distribution $q_\phi(z|x)$ fails to approximate the true posterior $p_\theta(z|x)$ and instead falls in a trivial solution of a one-to-one map between p_θ and q_ϕ, like a δ distribution . With the $\mu_i -> $ and $\sigma_i -> 0^+$, for each x, the $q_\phi(z|x_i)$ learns a distribution with disjoint support leading to the undesirable trivial solution. This limitation has been demonstrated using a simple example of fitting a mixture of Gaussian in [35].

2.1.4 Latest trends

In the VAE literature, many approaches [24, 14, 9, 32, 5] has been proposed which attempt to solve one or more of these problems. However, many of these are limited in scope or cause additional overhead. A relatively new approach [35] solves this problem by defining an objective function which replaces the divergence term $D_{KL}(q(z|x)||p(z))$ with the $D(q(z|x)||p(z))$. The new term considers a divergence between the prior distribution $p(z)$ and the marginal inference distribution $q_\phi(z)$. The new objective function is $L_{InfoVAE} = -\lambda D(q(z|x)||p(z)) + \mathbb{E}_{q_\phi}(z|x)$ and is called **InfoVAE** due to its symmetry with the InfoGAN objective function [8]. The family of models support any divergence, leading to different models with different tradeoffs. For instance, we can use adversarial divergence (leading to Adversarial AutoEncoders) [24], which uses Jenson Shannon divergence. In general, we can use any f-divergence. Other divergences considered in [35] are stein variational gradient [23] and maximum mean discrepancy [13].

In general, VAE allows us to perform efficient inference but the generated samples are not crisp. GAN [11] solves this problem by learning the density implicitly. We now discuss this framework in more detail.

3 Generative Adversarial Networks (GAN)

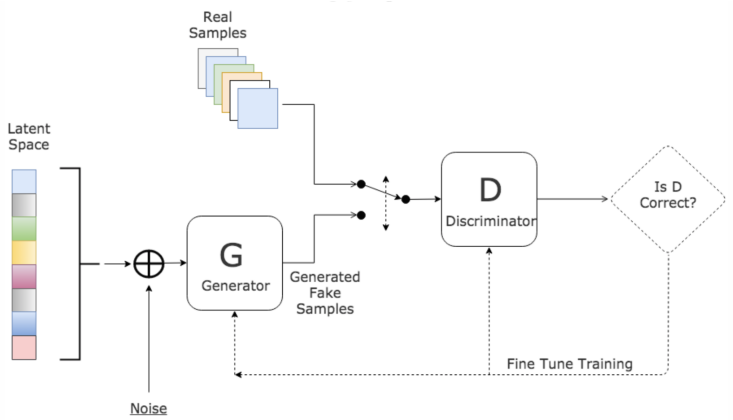

Figure 2: GAN overview[11]

GAN is trained as an adversarial game between two agents known as generator and discriminator. In this game, the job of the discriminator is to evaluate the samples as being fake (generated by the generator) or being real (taken from the true distribution) and the job of the generator is to fool the discriminator by generating real-looking samples. There are many variants of GAN, some of which we discuss in the following sections. [11] presents the first adversarial framework to represent the generator and the discriminator as multi-layer perceptron networks. The generator learns the data distribution p_g over data x using a network $G(z; \theta_g)$ and a prior over the input noise noise $p(z)$. The discriminator network $D(\theta_d)$ outputs a number $[0, 1]$ representing the probability of the sample being real or fake. This combined network is end-to-end differentiable. The discriminator tries to maximize $log\, D(x)$ for true samples and maximize $log(1 - D(z))$ for the generated samples. The generator is simultaneously trained to minimize the latter term. Here is the joint objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(z))]. \quad (2)$$

Maximizing likelihood objective can be roughly seen as $KL[p_{data}||p_g]$ does not work well due to its tendency to over-generalize. An alternative objective is $KL[p_g||p_{data}]$ which is more conservative and tends to not generalize well. The GAN minimizes the combination of both these objectives, which is known as the Jenson-Shannon divergence.

$$JSD[p_{data}||p_g] = JSD[p_g||p_{data}] = \frac{1}{2}KL\Big[p_{data}||\frac{p_{data}+p_g}{2}\Big] + \frac{1}{2}KL\Big[p_g||\frac{p_{data}+p_g}{2}\Big] \quad (3)$$

JS divergence minimised by GAN can also be interpreted as mutual information where GAN is a generative model augmented with an auxilary discriminator network. However, variational lower bound used for this optimization is not fully justified because using lower bound for a minimization problem isn't a good idea. [8] extends this information-theoretic framework with **InfoGAN** where an additional regularization term is added to the objective to encourage high mutual information between the generated samples and a subset of latent variables. This is again done by using a lower variation bound but this time it is used for maximization. The intuition is to encourage the subset of latent variables to represent the most salient sources of variance in the data as that would allow the model to learn disentangled representations in completely unsupervised fashion.

[34] views GAN as an energy-based framework where the discriminator is viewed as an energy function that tries to assign low energy values to the regions of high data density and higher values to other regions. It also uses a discriminator with an autoencoder architecture in which reconstruction error is the energy. Using reconstruction loss is expected to produce different gradient directions within the mini-batch and should allow for larger batch size as compared to the conventional binary logistic loss. However, they do not discuss the effect of choosing different objective functions on the behaviour of the algorithm. We believe that interpreting GAN in these frameworks is a step in the right direction of probabilistic modelling that would allow a better insight into its working and solving the challenges faced during GAN training.

3.1 Challenges and Solutions

GAN can be interpreted as a min-max game between the generator and the discriminator where they are trained in an alternating fashion. As per game theoretic perspective, training GAN requires finding the Nash equilibrium between the generator and the discriminator. This is a very difficult task as the cost function is non-convex and high dimensional. Thus, when GANs are trained to find a low-value of the cost function using gradient descent approach, it may fail to converge to Nash Equilibrium. Nash equilibrium occurs when the objective function of the discriminator is minimum with respect to the parameters of the discriminator and the objective function of the generator is minimum with respect to the parameters of the generator. However, minimizing one cost function can increase the other cost function and thus making convergence difficult.

[30] introduces some techniques to improve the convergence of GANs. The first trick is the feature matching that tries to address the instability by trying to force the generator to match the expected values of features for an intermediate layer of the discriminator. In doing so, we force the generator to match the features that discriminate most between the real and the generated data. Another main problem is mode collapse when the generator starts to output the same sample. Since the discriminator looks at each example separately thus gradients do not push the generator to produce dissimilar outputs. Once the network encounters the mode collapse, gradients are unable to revive the generator.[30] proposes mini-batch discrimination to help avoid mode collapse by enabling the discriminator to look at the combination of multiple examples. It can be seen as another input added for the discriminator that has the information of the distance between a particular sample in the batch and the other samples. It has been found to work better than the feature matching for generating visually appealing samples quickly. [1] and [31] also mention that the generative distribution of GAN is typically degenerate and the divergences that are tried to minimize are not well defined. They suggest that adding noise to both real and synthetic data can help in alleviating this problem. Another very related technique found to be helpful is one-sided label smoothing suggested by [30]. It

improves the training by strengthening the gradients for the generator by smoothing the target for the discriminator from 1 to 0.9. The common theme in these two tricks is to somehow cripple the discriminator which results in variational lower bound being not tight and allows better optimization when the support of the generator distribution has no overlap with the true data distribution. All of these techniques are empirically shown to help with the GAN training but no or very less theoretical reasoning is provided.

Some papers have explored different cost functions to stabilize GAN training. One such example is [2] that proposed **WGAN** that focuses on optimizing an approximation of the EM distance (Wasserstein-1). EM distance can be loosely defined as the minimal work required to shift mass of one distribution to change it into another distribution. It prevents the need of balancing the discriminator and the generator distance carefully. It also reduces the mode collapse problem seen in typical GAN setting. The EM distance can be continuously estimated by training the discriminator to optimality. The paper provides strong theoretical reasoning justifying the advantage of WGAN over GAN. It also evaluates other distances like Total Variation, KL Divergence and JS Divergence to conclude that EM distance is the most suitable choice as it offers better guarantees of continuity and differentiability of the loss function. The correlation between the discriminator loss and the perceptual quality of the image is also observed and WGAN samples are found to be more detailed. One notable difference in WGAN and GAN setting is in the training of discriminator. The discriminator of WGAN is trained to convergence for better flow of gradients to the generator. It isn't so clear that how would the results change if the alternating training is tried in WGAN and the discriminator isn't trained to convergence. WGAN has also been found less sensitive to batch normalization and the choice of non-linearities.

Some papers have also pointed towards the orthogonal direction of this problem. Optimization techniques for finding Nash-equilibrium during GAN training are found to be inadequate for the convergence. [25] shows that simultaneous gradient ascent used for finding local Nash-equilibria may fail to converge due to the presence of eigenvalues of the Jacobian of the associated gradient vector field with zero real-part and eigenvalues with a large imaginary part. They provide a solution that tries to make the associated gradient field vector well behaved by combining it with a conservative field vector. Similarly, [15] suggests updating the generator and discriminator at different rates helps the algorithm to converge to local Nash-equilibrium. [27] draws a connection between adversarial training and reinforcement learning. It views GANs as actor-critic methods in an environment where the actor cannot affect the reward. Both are known to be difficult to optimize.

3.2 Applications

GANs are extensively used in various generative and transformation tasks. One such example is [36] that uses GAN for image transformation by learning a mapping between the images of source domain to the images of the target domain and vice-versa. To make this work an additional term of cycle loss is added to ensure that mappings do not contradict each other. This additional term tries to enforce the reconstruction of the source image after a cycle of image transformations. [36] demonstrates its applicability using collection style transfer, generating season transfer photos and photos from pictures, However, although the results are impressive, they are not uniformly positive. Also, there is a huge performance gap between this model and the supervised models trained using paired training data.

Another interesting application of GANs is text to image synthesis which involves generating an image conditioned on the given natural language text description. [28] introduces GAN based end-to-end differentiable architecture which learns a joint embedding for text and image features. They also introduce a third type of input consisting real images with mismatched text as fake samples to enforce the idea of a matching-aware discriminator. Manifold interpolation term is added to the generator objective to fill in the gaps between the training points on the data manifold. The model works pretty well on the dataset with images and flowers, however, the generated samples lack details and when applied to MS-COCO image dataset, it is found that generated scenes are generally not coherent.

[10] introduces a GAN based approach to do the reverse task of generating captions based on given text. It tries to learn a generator for image captioning and a discriminator that can distinguish between the generated captions from those present in the data-set. It is different from other MLE-based captioning models as the captions produced by this model are more human-like and diverse than the counterpart. GANs are currently gaining popularity in other fields like drug discovery[4] and molecule development in oncology[19].

4 Discussion

4.1 Combining GAN and VAE

[21] combines GAN and VAE to get the best of both the worlds. It is known that the discriminator network of GAN can implicitly learn a rich similarity metric for images, so as to discriminate them from fake samples. They exploit this observation so as to transfer the properties of images learned by the discriminator into a more abstract reconstruction error for the VAE. With the learned distance measure, they are able to train VAE network generating images of unprecedented visual quality. [18] presents a new formulation that connects GAN and VAE under a unified statistical view linking them to classic wake-sleep algorithm[17]. The key idea here is to interpret GAN as performing inference and discriminator as a generative process that produces real/ fake labels. The unified view is also useful to analyze existing model variants, and aids to exchange ideas across research lines in a principled way. They demonstrate this by transferring the importance weighting method in VAE literature for improved GAN learning, and an adversarial mechanism to enhance VAEs. A new method, adversarial autoencoder [24] combines GANs and VAEs to perform efficient variational inference. In this framework the autoencoder is trained using dual objective: the traditional reconstruction error criterion and an adversarial criterion. The new criterion provides an additional regularization by using a GAN to match the aggregated posterior $p(z)$ of the hidden code to an arbitrary prior distribution. This ensures that sampling from any part of the prior generates meaningful samples.

4.2 Conclusion

Until recently, deep generative models were hard to train due to inaccurate and inefficient MCMC methods. GANs and VAEs avoid these difficulties by training directly using backpropagation. Both of these models have produced significant breakthroughs in the field of unsupervised training. GANs are best for generating crisp samples, while VAEs focus on representing the latent space effectively. Significant research is going on to extend these models to new domains and make them more accurate. Learning generic and effective representations is an important field of research in VAEs. However, there are challenges like overfitting to supervised tasks, entanglement of codes, and uninformed codes, etc. Additionally, often VAEs fail to produce crisp and clear samples. This has been attributed to the less expressive Gaussian conditional density assumption. Using a more powerful density function, like an autoregressive network, is seen as another potential area of research. There have also been efforts to merge VAEs and generic probabilistic graphical models to make them more powerful. However, efficient inference in these models remains an open problem. Solving the stability issues of GAN remains a high priority research area for the future. The research community has proposed several tricks and techniques to stabilize their training. However, most of these techniques are not backed up by strong theoretical reasoning and have not been exhaustively tested. As discussed before, we cannot take optimization techniques used for training the conventional networks as granted for GAN-like frameworks. Therefore, we believe improving the optimization methods for these frameworks would be another significant thread of research in near future. There have been numerous attempts to understand GANs and the stability issues in different theoretical frameworks but none of these frameworks so far can be solely credited for providing sufficient interpretability. There have also been attempts to unify GAN with other models like VAE. In future, we expect to see more of such frameworks that would unify the current understanding of these models with other domains and would allow the exchange of ideas across the research lines in a more principled way. Lastly, there lies a big opportunity to apply adversarial training framework to a plethora of other generative tasks that involve high-dimensional and multi-modal distributions.

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] M. Benhenda. Chemgan challenge for drug discovery: can ai reproduce natural chemical diversity? *arXiv preprint arXiv:1708.08227*, 2017.

[5] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

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

[7] W. Chen and J. Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. *arXiv preprint arXiv:1801.02753*, 2018.

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[9] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[10] B. Dai, S. Fidler, R. Urtasun, and D. Lin. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, and D. Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pages 3549–3557, 2016.

[13] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[14] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

[15] 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. In *Advances in Neural Information Processing Systems*, pages 6629–6640, 2017.

[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[17] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[18] Z. Hu, Z. Yang, R. Salakhutdinov, and E. P. Xing. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017.

[19] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, and A. Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883, 2017.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[22] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

[23] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

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

[25] L. Mescheder, S. Nowozin, and A. Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems*, pages 1823–1833, 2017.

[26] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[27] D. Pfau and O. Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016.

[28] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[29] K. Ridgeway. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299*, 2016.

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[31] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[32] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.

[33] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[34] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[35] S. Zhao, J. Song, and S. Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.