Mode Collapse In Generative Adversarial Networks: A Survey

Diplav
diplav@g.ucla.edu

Gaurav Singh
grvsingh@g.ucla.edu

Srinath Naik Ajmeera
srinath@cs.ucla.edu

Abstract

Though Generative Adversarial Networks are widely successful, there are some inherent problems such as non-convergence and mode collapse which limit its usage. In this paper, we focus on mode collapse in detail, address why it is a bottleneck on effectiveness and usage of GANs, discuss and summarize the different strategies in the literature followed by identifying open research directions.

1 Introduction

Generative models have progressed a long way into the history of machine learning. Broadly, these can be classified into explicit models where a potential probability distribution of the samples is calculated and implicit models in which the interaction is only through sampling. Early models like Fully Visible Belief Networks, Independent Component Analysis use maximum likelihood estimation, markov chain monte carlo techniques to estimate the probability distribution, but these have difficulties in defining a tractable density function and needed heavy computational power. Later, deep learning based implicit models like Variational Auto Encoders[1][2], Generative Stochastic Networks[3] are developed, which offer advantages of direct sample generation and simpler training process. However, they are not able to generate sharp examples and often do not map the input data distribution accurately.

The introduction of Generative Adversarial Networks[4] have arose significant research in this area due to added advantages like parallel generation, minimum restrictions, no dependency on markov chains and unprecedented practical effectiveness. Variations of GANs (DCGAN[5], StackGAN[6] etc.) were constructed for a plethora of use cases by using appropriate loss functions and architectural changes. Although GANs theoretically achieve Nash equilibrium and capable of learning any probability distribution, their practical implementation suffers from issues such as non-convergence, vanishing gradients and mode collapse.

Mode collapse refers to the situation where a GAN is able to generate only few modes(variations) of the input data distribution even though input data contains many modes. This issue in GANs has been in existence since their introduction and is a substantial hindrance to many practical applications. Solving mode collapse is a challenging problem and several efforts were made in regards to it. In this paper, we present a comprehensive survey of various methodologies proposed over time to tackle this problem. We describe the methods, their theoretical basis, potential drawbacks and adaptability. At the end, we discuss current progress related to occurrence and mitigation of mode collapse including some possible future directions of research.

2 Generative Adversarial Networks

Adversarial networks are modelled as a zero-sum game between a discriminator D and a generator G. The generator maps elements from a latent distribution p_z to the instance space X, there by creating a generator distribution p_g, where as the discriminator's task is to classify the input variable x as either from real distribution p_{data} or a generated one (p_g). The value function is as follows.

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

Where $D(x)$ represents the probability that given input x is from real distribution p_{data} rather than from p_g. The discriminator tries to maximise this value where as the generator tries to minimize it, there by following a mini-max game strategy. To find the nash equilibrium of this game, the optimal discriminator D^* for any given generator G is found as follows.

$$\begin{aligned} V(G,D) &= \int_x p_{data}(x)log(D(x))dx + \int_z p_z(z)log(1-D(G(z)))dz \\ &= \int_x p_{data}(x)log(D(x)) + p_g(x)log(1-D(x))dx \end{aligned}$$

The above equation is concave with respect to D for $D(x) \in [0,1]$ and achieves maximum value at

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$$

Substituting $D^*_G(x)$ in Equation (1), we can rewrite the value function for G as follows.

$$\begin{aligned} C(G) &= \mathbb{E}_{x\sim p_{data}(x)}[logD^*_G(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D^*_G(G(z)))] \\ &= \mathbb{E}_{x\sim p_{data}(x)}[logD^*_G(x)] + \mathbb{E}_{x\sim p_g(x)}[log(1-D^*_G(x))] \\ &= \int_x p_{data}(x)\left[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + \int_x p_g(x)\left[log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right] \\ &= -log4 + KL\left(p_{data}\middle|\middle|\frac{p_{data}+p_g}{2}\right) + KL\left(p_g\middle|\middle|\frac{p_{data}+p_g}{2}\right) \\ &= -log4 + 2\cdot JSD(p_{data}||p_g) \end{aligned}$$

Where KL is Kullback-Leibler divergence and JSD is the Jensen-Shannon divergence. JSD between two distributions is non-negative and achieves its minimum 0 only when $p_g = p_{data}$. This proves the convergence and ability of adversarial nets to map any input distribution.

Assuming $G(z;\theta_g)$ and $D(x;\theta_d)$ are both parametrized differentiable functions, this equilibrium can be achieved by simultaneous gradient descent(ascent). D is updated using gradient ascent for k steps to achieve the optimal and then G is updated once using gradient descent. This process is repeated until convergence. Refer to [4] for detailed algorithm.

Conditional GANs[7] extend GAN by providing a conditioning on the input latent variable z and also the input x to generate conditioned output to capture different modes of the input data. Various GANs were developed eventually by using different loss functions, architectural and algorithmic changes.

3 Mode Collapse

When the elements in latent space map to a single or very few concise elements in the instance space, the generator is unable to capture most of the variations in the input distribution, which is termed as mode collapse. Fig. 1 provides an intuition of such a scenario. During the training phase, it might happen that the generator learns to produce a single set of outcomes for which the discriminator fails to identify, there by giving the same output each time. Eventually, the discriminator learns to classify this set, however generator might shift its strategy to generate some other mode, and hence creates a cycle between G and D, each shifting between generation and discrimination of various modes of the input distribution.

A complete collapse is uncommon in general but partial collapse occurs often. The fundamental reason for occurrence of mode collapse is attributed to the following.

- $G(z;\theta_g)$ and $D(x;\theta_d)$ are parametric approximations and therefore the concavity and convexity properties of Equation (1) do not hold anymore, hence convergence is not guaranteed.
- Gradient descent based training do not differentiate between mini-max and maxi-min and there by sometimes achieve $max_D\, min_G\, V(G,D)$ instead of $min_G\, max_D\, V(G,D)$.

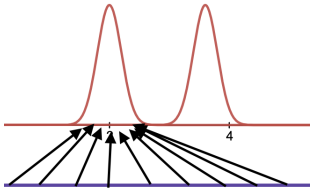

Figure 1: Mode collapse example

To give some practical insights, we describe two examples which suffer from mode collapse. First one is a constructed problem mentioned in [8] where the target distribution consists a mixture of 8 different gaussians. As we can see from Fig. 2 that while training on this dataset, the generator shifts between various modes and never achieves the expected distribution. Next one is a text-to-image generation task from [9] where the generator creates visually similar images for varied input noise, conditioned on the same features(input vector of text here), as evident from Fig. 3.

Figure 2: An illustration of the mode collapse problem on a two-dimensional toy dataset. Images from [8]

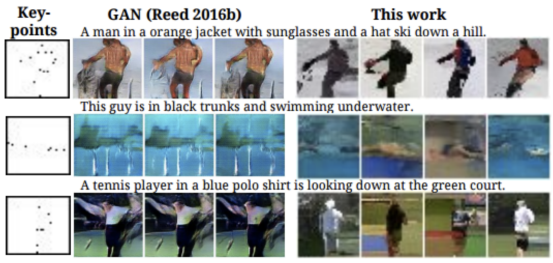

Figure 3: An illustration of the mode collapse problem on a text-to-image task. Images from [9]

Initially, mode collapse was thought to occur due to limited generator capacity and the kind of divergence that GANs are minimizing, which is similar to $KL(p_g||p_{data})$. Where as methods like MLE minimise $KL(p_{data}||p_g)$ divergence which can output many modes at the cost of poor sample generation. Fig. 4 from [8] shows this graphically for a generator capable of producing only one gaussian where as the target distribution is a mixture of two gaussians. This idea is abandoned later as GANs were developed to minimize various kind of divergences and still mode collapse is observed in them.

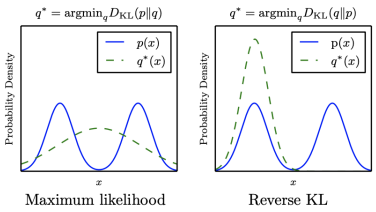

Figure 4: Mode collapse reasoning based on divergence. Images from [8]

4 Mitigation Strategies

4.1 Early methods

[10] proposes minibatch features where the discriminator compares an example to a minibatch of generated data, by using the distance between the example and generated samples in latent space, it is able to understand if the example is predominantly similar to the generated batch and propagate appropriate additional weight to the generator loss. Although this method works well, it is extremely sensitive to the definition of features and is not scalable for large scale applications with various modalities. [11] defines unrolled GANs where the discriminator is updated k steps to get a surrogate loss function $f_K(\theta_G, \theta_D)$ and the generator is one step updated based on gradient of this surrogate loss. The discriminator is then rolled back of the performed k step updates to finally have only single step update w.r.t $f(\theta_G, \theta_D)$. Respective equations to calculate $f_K(\theta_G, \theta_D)$ and final updates are summarized below.

$$\theta_D^0 = \theta_D$$

$$\theta_D^{k+1} = \theta_D^k + \eta^k \frac{df(\theta_G, \theta_D^k)}{d\theta_D^k}$$

$$f_k(\theta_G, \theta_D) = f(\theta_G, \theta_D^k(\theta_G, \theta_D))$$

$$\theta_G \leftarrow \theta_G - \eta \frac{df_K(\theta_G, \theta_D)}{d\theta_G}$$

$$\theta_D \leftarrow \theta_D + \eta \frac{df(\theta_G, \theta_D)}{d\theta_D}$$

The idea here is that the generator is able to perceive future updates of discriminator to get an idea of where it is heading to change its strategy. Drawbacks of this method include added computational complexity, tuning for k and scaling to larger tasks.

4.2 Using Regularization

Mao et al. [12] have proposed to add a mode seeking regularization term in the main loss to mitigate the problem of mode collapse. Mode seeking GANs [12] basically tries to tackle the problem of mode collapse in conditional GANS. CGANS are built upon GANS and take contextual information as an additional input. CGANs are naturally multimodal in the sense that the same input can correspond to many possible outputs. The problem of mode collapse is much more severe in the case of CGANs.

The simplest possible approach to deal with model collapse is to add noise along with contextual information as input where contextual information will provide the content and noise will add variation to the output. But this is not so simple in the case of CGANs as the dimension of contextual information is generally much higher as compared to the random noise vector added, so GANs pay more attention to learning from high dimensional contextual information and the variation because of noise is generally neglected. Most of the recent approaches to deal with mode collapse are computationally heavy and involve additional auxiliary networks which are task-specific and can't be generalized to a different framework.

This work is based on the observation that mode collapse occurs when two latent codes, z_1 and z_2, are close to each other in the latent space and the corresponding generated images, $G(z_1)$ and $G(z_2)$, are mapped to the same mode.

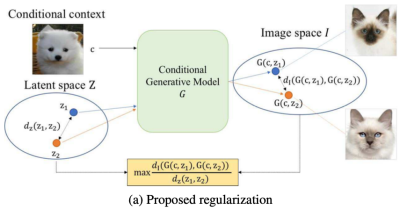

Figure 5: Proposed Algorithm [12]

$$L_{ms} = \max_{\mathrm{G}}(\frac{d_I(G(z_1), G(z_2))}{d_Z(z_1, z_2)})$$

The mode seeking regularization term L_{ms} seeks to maximize the ratio of the distance between the latent code z and their corresponding generated image G(z) which adds a minimal computational overhead and can be generalized to a different framework.

As a consequence of which generator is now forced to explore the different modes in target distribution and generate results that belong to different modes for two latent codes which are close to each other in latent space and the discriminator now gets trained with dissimilar images and gradient even from minor modes gets propagated, which were ignored earlier.

$$L_{new} = L_{original} + \lambda_{ms} L_{ms}$$

4.3 Using Implicit Variational Learning

Akash et .al [13] used an auxiliary network to avoid model collapse. The main idea of VEEGAN [13] is to introduce an additional reconstructor network F_θ that maps the true data distribution p(x) to gaussian random noise. Both the generator and reconstructor are trained simultaneously which enforces the reconstructor not only to learn to map true data distribution to a gaussian but also learn to approximately reverse the action of the generator G_γ. The main motivation behind this approach is if the reconstructor can learn to both, approximate inverse of the generator and to map the true data distribution to gaussian then it will enforce the generator to map the noise gaussian distribution to the entirety of true data distribution. Intuitively if the same reconstructor maps both generator and true distribution to gaussian then the generator distribution will most likely coincide with the true distribution.

To assess whether F is an approximate inverse of G we compute L2 norm between $z \in p_0$ and $F_\theta(G_\gamma(z)))$. To measure whether F_θ maps true data distribution p(x) to gaussian we use cross-entropy loss between Z and $F_\theta(x)$.

$$O_{entropy}(\gamma, \theta) = \mathbb{E}[||z - F_\theta(G_\gamma(z)))||_2^2] + H(Z, F_\theta(X))$$

However this objective function is not computable. The output of the reconstructor when applied to $X \sim p(x)$ can be written as $p_\theta(z) = \int_x p_\theta(z|x)p(x)dx$. Thus $H(Z, F_\theta(X))$ can written as

$$H(Z, F_\theta(X)) = -\int p_o(z)p_\theta(z)dz = \int p_o(z)log \int_x p_\theta(z|x)p(x)dxdz$$

To make this term computable, we introduce a variational distribution $q_\gamma(x|z)$ and by Jensen's inequality we can approximate it to

$$-logp_\theta(z) = -log \int_x p_\theta(z|x)p(x)\frac{q_\gamma(x|z)}{q_\gamma(x|z)}dx \leq -\int q_\gamma(x|z)log\frac{p_\theta(z|x)p(x)}{q_\gamma(x|z)}$$

$$-\int p_o(z)logp_\theta(z) \leq \mathbb{KL}[q_\gamma(x|z)p_o(z)||p_\theta(z|x)p(x)] - \mathbb{E}[logp_o(z)]$$

From above equation the objective function reduces to

$$O(\gamma, \theta) = \mathbb{E}[||z - F_\theta(G_\gamma(z)))||_2^2] + \mathbb{KL}[q_\gamma(x|z)p_o(z)||p_\theta(z|x)p(x)] - \mathbb{E}[logp_o(z)]$$

Rather than minimizing $O_{entropy}(\gamma, \theta)$, we minimize a surrogate loss function $\mathbb{O}$ However we can't directly optimize $\mathbb{O}$ as it depends on the KL divergence of density ratio which is unknown, since q_γ is implicit and $p(x)$ is unknown. The authors have determined this density ratio using a discriminator which is trained to follow

$$\mathbb{D}_\omega(z, x) = log\frac{q_\gamma(x|z)p_o(z)}{p_\theta(z|x)p(x)}$$

Hence $\mathbb{O}$ is determined using the equation

$$\hat{\mathbb{O}}(\omega, \gamma, \theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{D}_\omega(z^i, x_g^i) + \frac{1}{N}\sum_{i=1}^{N}d_\omega(z^i, x_g^i)$$

where $(z^i, x_g^i) \sim p_o(z)q_\gamma(x|z)$ and $\mathbb{D}_\omega$ is trained to distinguish data from $q_\gamma(x|z)p_o(z)$ and $p_\theta(z|x)p(x)$ using the objective function:

$$\mathbb{O}_{LR}(\omega, \gamma, \theta) = -\mathbb{E}_\gamma[log(\sigma(D_\omega(z, x)))] - \mathbb{E}_\theta[log(1 - \sigma(D_\omega(z, x)))]$$

where E_γ is expectation w.r.t $q_\gamma(x|z)p_o(z)$ and $\mathbb{E}_\theta$ is expectation w.r.t to $p_\theta(z|x)p(x)$.

4.4 Using additional auxiliary network

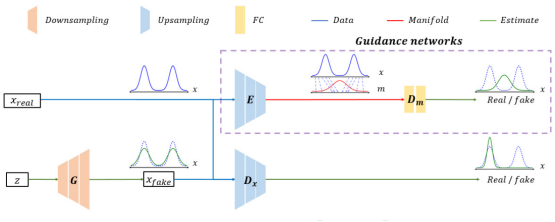

Figure 6: Proposed model [14]

Bang et al. [14] proposed MGGAN which uses an auxiliary network to tackle with mode collapse. The role of the guidance network here is to induce the generator so that P_{model} matches P_{data} in manifold space. The guidance network consists of an encoder E and discriminator D_m. The encoder maps P_{data} and P_{model} in manifold space P_m and $\hat{P_m}$, respectively while the discriminator assesses the dissimilarity between P_{model} and P_{data} in the manifold space. The encoder of the guidance network is trained independently while the discriminator of the guidance network is trained jointly with the discriminator and generator of the GAN.

The encoder of the guidance network is designed such that it represents the meaningful representation of data in manifold space and all the modes of the data p(x) are covered by the output of encoder P_m in the manifold space. The encoder of the pre-trained autoencoder is used since it satisfies all these conditions. As the encoder of the autoencoder is trained to minimize the reconstruction error it generally captures all the modes of true data distribution. The encoder of the guidance is pretrained using true data and is fixed.

The discriminator of the guidance network learns to evaluate the dissimilarity between the encoded output of P_{data} and P_{model} i.e P_m and $\hat{P_m}$ in the manifold space. By changing the weights of the two-loss term we can control the strength of the guidance network thus affecting the degree of diversity of the generator network.

$$\min_{\mathrm{D_x},\mathrm{D_m}} \mathbb{E}_{x\sim P_{data}}[log(D_x(x)) + log(D_m(E(x)))]+$$
$$\mathbb{E}_{z\sim P_Z}[log(1 - D_x(G(z))) + log(1 - D_m(E(G(z))))]$$
$$\min_{\mathrm{G}} - \mathbb{E}_{z\sim P_Z}[\alpha log(D_x(G(z))) + (1-\alpha) log(D_m(E(G(z))))]$$

The two discriminators don't explicitly affect the output of each other, although both of them individually try to modify the output of the generator. The generator tries to accomplish two goals at once, first is to minimize the difference between P_{data} and P_{model} just as in case of vanilla GAN, and the second is to minimize the difference between the encoded output of P_{data} and P_{model} i.e. P_m and $\hat{P_m}$ in manifold space.

4.5 Using multiple generators

Li, Wei, et al [15] proposes the use of orthogonal value as a true measure of how much distinct two information vectors are. Using multiple generators (MGAN) [16] to tackle the problem of Mode collapse, has proven to be quite consistent but the "shared parameters" method to train all generators together makes the generator learns the same overlapping modes. The underlying problem is that although the capacity of the architecture is increased to learn new modes, there is no barrier to stop them from learning the modes learnt by other generators. MGO-GAN which stands for Multi Generator Orthogonal GAN proposed an architecture as shown below, with K generators $G_{1:K}$, one pre-trained Encoder (E) to abstract the feature vectors from generated output $G(z)_i$ given random noise z as input and one Discriminator D same as vanilla GAN which discriminates the real and generated data. The encoder used by the paper is a Conv- BatchNorm-LeakyReLu feed forward Neural Network.

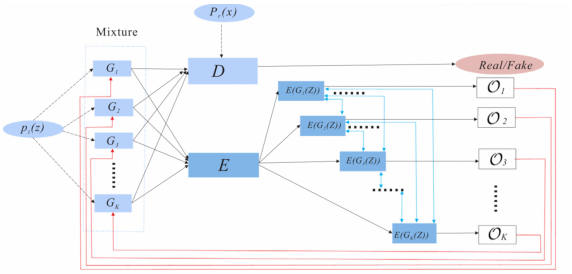

Figure 7: Architecture of MGO-GAN [15]

Orthogonal values are computed as inner products among different pairs of the encoded information $E(G_i(z))$ and is represented as,

$$O(E(G_1(z)), E(G_2(z))) = \frac{|(E(G_1(z)), E(G_2(z)))|}{|E(G_1(z))||E(G_2(z))|}$$

Orthogonal values loyally reflects the correlation between the information learned by the generators and this is integrated in the Generator's training criterion and has to be minimized during training when MGO-GAN plays the mini-max game. Lower orthogonal value among generated information means less similarity among fake generated data and hence much more possibility of achieving different modes from different generators. The discriminator's optimization task is same as the vanilla GAN (cite). Given the real probability distribution as p_r, and initial input noisy distribution as p_z, the minimization and maximization tasks corresponding to Generator (G) and Discriminator (D) respectively are,

$$\min_{G_{1:K}} \quad V(G_{1:K}, D) = \mathbb{E}_{z\sim p_z(z)}[log(1 - D(G_i(z)))] + \frac{1}{2}\sum_{i\neq j} \lambda O(E(G_i(z)), E(G_j(z)))$$

$$\max_{D} \quad V(G_{1:K}, D) = \mathbb{E}_{x\sim p_r(x)}[logD(x)] + \frac{1}{K}\sum_{i=1}^{K} \mathbb{E}_{z\sim p_z(z)}[log(1 - D(G_i(z)))]$$

Each generator G_i implicitly defines a probability distribution p_{G_i} as the distribution of generated samples $G_i(z)$. Altogether, such K generators approximate the real distribution p_r. It is shown in the paper that for $G_{1:K}$ fixed the optimal Discriminator D^* takes the form of,

$$D^*(x) = \frac{p_r}{p_r + \frac{p_{G_1} + p_{G_2} + ... + p_{G_K}}{K}}.$$

The above result is analogous to the result of optimal Discriminator D^*_G from original GAN [4] paper as shown above, where $p_{data}(x)$ is same as p_r and $p_g(x)$ is analogous to average of K generated probability distributions p_{G_i}. To get the above result, the optimization task of Discriminator D given by $V(G_{1:K})$ is first expanded in terms of probabilities of random variables x and z and then by finding the partial derivative of $\frac{\partial V(D(x))}{\partial D(x)}$ to get $\frac{p_r}{D(x)}$ so as to maximize D in the range [0,1].

Another result shows that the global minimum for the training criterion of Generator $V(G)$ is achieved if and only if, $p_r = \frac{p_{G_1} + p_{G_2} + ... + p_{G_K}}{K}$ and each generator captures information which others are not able to capture i.e. the total orthogonal value of the architecture which is the sum of orthogonal value between all $E(G_i(z))$ pairs is minimized.

The paper discusses about the convergence of the proposed mini-max formulation and shows that, assuming all generators induce the generator model distribution $p_m = \frac{p_{G_1} + p_{G_2} + ... + p_{G_K}}{K}$ and given D has enough capacity and is at optimal status w.r.t all generators $G_{1:K}$, then p_m converges to p_r.

Multiple generator based approaches to tackle Mode Collapse provides significant improvements but there are some drawbacks to it. Architectures with multiple generators are quite computationally expensive to train. Multiple generator architectures usually require some auxiliary networks such as Encoders whose reliability to extract important features can be a bottleneck to overall architecture's performance also it is not guaranteed that encoders will map the output of Generator network to the same comparable feature space.

5 Discussion

GANs suffer from problem of mode collapse. The mitigation strategies can be broadly classified in two main categories, based on Architectural changes and based on Objective function changes.

Approaches based on Architectural changes involves an additional auxiliary network along with GAN architecture as seen in MGO-GAN where multiple generators along with a metric (orthogonal value) to measure the similarity in generated data, or ADA-GAN [17] where multiple weak generators are used, which are able to perform well with some modes to get strong generator network. VEE-GAN and MG-GAN uses an additional auxiliary encoder architecture to map $p(G(z))$ and $p(x)$ into a lower dimensional space to penalize the mode imbalance in GAN training.

Approaches based on Objective function changes involves a change in the optimisation function of the Generator or Discriminator network such as WGAN [18] which uses Earth Mover's distance as GAN loss function and a measure of difference between generated data and real data distributions instead of JS divergence. Mode Seeking GAN which uses a mode seeking regularization term to maximize the ratio of the distance between the latent code and their corresponding generated images.

The potential areas of research identified while doing this survey are that no good metric exists which can measure the efficacy of mode collapse mitigation strategies or to evaluate different GAN architectures in general. Also one open research direction is to focus on changing the discriminator's capabilities. We have seen that most of the strategies proposed to tackle mode collapse revolve around changing generators but to think of it Mode collapse is contributed by Discriminator as well as building a discriminator which can assess the diversity of data generated is still an open research area.

6 Conclusion

Overall, we explored a variety of techniques to mitigate mode collapse problem, possible reasons for its occurrence and discussed open research directions in this area.

Acknowledgements

Thanks to the class of students in Machine Learning Algorithms Fall 2021, the TA's and Prof. Quanquan Gu for such a wonderful course.

References

[1] Diederik P Kingma. "Fast gradient-based inference with continuous latent variable models in auxiliary form". In: *arXiv preprint arXiv:1306.0733* (2013).

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *International conference on machine learning*. PMLR. 2014, pp. 1278–1286.

[3] Yoshua Bengio et al. "Deep generative stochastic networks trainable by backprop". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 226–234.

[4] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[5] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[6] Han Zhang et al. "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5907–5915.

[7] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[8] Ian Goodfellow. "Nips 2016 tutorial: Generative adversarial networks". In: *arXiv preprint arXiv:1701.00160* (2016).

[9] Scott Reed et al. "Generative adversarial text to image synthesis". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1060–1069.

[10] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016), pp. 2234–2242.

[11] Luke Metz et al. "Unrolled generative adversarial networks". In: *arXiv preprint arXiv:1611.02163* (2016).

[12] Qi Mao et al. "Mode seeking generative adversarial networks for diverse image synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1429–1437.

[13] Akash Srivastava et al. "Veegan: Reducing mode collapse in gans using implicit variational learning". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 3310–3320.

[14] Duhyeon Bang and Hyunjung Shim. "MGGAN: Solving Mode Collapse Using Manifold-Guided Training". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2347–2356.

[15] Wei Li et al. "Tackling mode collapse in multi-generator GANs with orthogonal vectors". In: *Pattern Recognition* 110 (2021), p. 107646.

[16] Quan Hoang et al. "MGAN: Training generative adversarial nets with multiple generators". In: *International conference on learning representations*. 2018.

[17] Ilya Tolstikhin et al. "Adagan: Boosting generative models". In: *arXiv preprint arXiv:1701.02386* (2017).

[18] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 214–223. URL: `https://proceedings.mlr.press/v70/arjovsky17a.html`.