Inception Score, Label Smoothing, Gradient Vanishing and $-\log(D_r(x))$ Alternative

Zhiming Zhou
Shanghai Jiao Tong University
heyohai@apex.sjtu.edu.cn

Weinan Zhang
Shanghai Jiao Tong University
wnzhang@sjtu.edu.cn

Jun Wang
University College London
j.wang@cs.ucl.ac.uk

Abstract

In this paper, we study several GAN related topics mathematically, including Inception score, label smoothing, gradient vanishing and the $-\log(D_r(x))$ alternative. We show that Inception score is actually equivalent to Mode score, both consisting of two entropy terms, which has the drawback of ignoring the prior distribution of the labels. We thus propose AM score as an alternative that leverages cross-entropy and takes the reference distribution into account. Empirical results indicate that AM score outperforms Inception score. We study label smoothing, gradient vanishing and $-\log(D_r(x))$ alternative from the perspective of class-aware gradient, with which we show the exact problems when applying label smoothing to fake samples along with the $\log(1-D_r(x))$ generator loss, which is previously unclear, and more importantly show that the problem does not exist when using the $-\log(D_r(x))$ generator loss.

1 Introduction

Generative adversarial nets (GANs) [9] as a new way of generating samples has recently drawn much attention. Generally, GAN consists of two distinct neural networks competing with each other: the generator network aims to generate samples in order to approximate the underlying data distribution, whereas the discriminator network aims to distinguish a sample as to whether it is real or generated from the generator.

Since its emergence in 2014, variants of GAN techniques have been proposed. Denton et al. [7] used cascade Laplacian Pyramid to generate images in a coarse-to-fine fashion. Radford et al. [25] proposed a class of deep convolution network structure which makes GAN training more stable. Larsen et al. [15] combined variational autoencoder [14] and GAN to provide better training. Che et al. [5] added mode regularizer and trained GAN in a manifold-diffusion fashion. Metz et al. [20] defined an alternative generator objective w.r.t. an unrolled optimization of the discriminator, trying to solve the minimax conflict and avoid mode collapse. Besides, variant alternative objectives are proposed, such as: Integral Probability Metrics [2, 10, 1, 21], Energy-Based Model [31, 6, 13], Least-Square GAN [19], f-GAN [24], BEGAN [3], Geometric GAN [18]. And GAN has been applied in various applications [23, 29, 30, 11, 16, 4, 12] and achieves great success.

However, we found several basic GAN concepts are not clearly stated in the previous literature, which results into common inaccurate understandings. In the paper, we study a portion of these topics mathematically to correct some potential misunderstandings or mistakes, including Inception score, label smoothing, gradient vanishing and the $-\log(D_r(x))$ alternative.

2 Inception Score and Reference Distribution

One of the difficult problems in generative models is how to evaluate them [28]. With the hypothesize that: in multiple exclusive classes setting, a good sample from G should be classified to one class by

D with a high confidence, i.e. with a sharp probability distribution over classes rather than a flat one or a weighted one, evaluation metric like Inception score [26] was proposed.

In this section, we first present two related existing metrics, i.e., the Inception score [26] and the MODE score [5], and point out their drawbacks when the training (reference) data class is not evenly distributed. Then we present the new AM score which solves such a problem, and we suggest using an accordingly pre-trained classifier for each dataset.

2.1 Inception Score

As a recently proposed metric for evaluating the performance of a generator, the Inception score is found to be well correlated with human evaluation [26], where a pre-trained publicly-available Inception model C is introduced. By applying the Inception model to each generated image x and getting the corresponding class probability distribution judged by C, i.e. $C(x)$, the Inception score is calculated via:

$$\text{Inception score} = \exp\Big(\mathbb{E}_x\big[\mathrm{KL}\big(C(x) \parallel \bar{C}^G\big)\big]\Big), \tag{1}$$

where $\mathbb{E}_x$ is the short of $\mathbb{E}_{x\sim G}$ and $\bar{C}^G = \mathbb{E}_x[C(x)]$ is the overall class probability distribution of the generated samples judged by C, and KL denotes the Kullback-Leibler divergence and is defined as:

$$\mathrm{KL}(p \parallel q) = \textstyle\sum_i p_i \log \frac{p_i}{q_i} = \sum_i p_i \log p_i - \sum_i p_i \log q_i = -H(p) + H(p,q).$$

A particular drawback of the Inception score is it does not take into account the prior distribution of the labels. An extended measure, the MODE score, is proposed in [5], which is calculated via:

$$\text{MODE score} = \exp\Big(\mathbb{E}_x\big[\mathrm{KL}\big(C(x) \parallel \bar{C}^{\text{train}}\big)\big] - \mathrm{KL}\big(\bar{C}^G \parallel \bar{C}^{\text{train}}\big)\Big), \tag{2}$$

where the overall class probability distribution $\bar{C}^{\text{train}}$ from the training data has been added as a reference point. However, the MODE score and the Inception score are, in fact, equivalent. To see it, we introduce the following lemma.

Lemma 1. *Let $p(x)$ be the class probability distribution of the sample x that from a certain data distribution, and $\bar{p}$ denote the reference probability distribution, then*

$$\mathbb{E}_x\big[H\big(p(x), \bar{p}\big)\big] = H\big(\mathbb{E}_x\big[p(x)\big], \bar{p}\big). \tag{3}$$

Proof.

$$\begin{aligned}
&\mathbb{E}_x\big[H\big(p(x), \bar{p}\big)\big] = \mathbb{E}_x\big[-\textstyle\sum_i p_i(x) \log \bar{p}_i\big] \\
&= -\textstyle\sum_i \mathbb{E}_x[p_i(x)] \log \bar{p}_i = -\sum_i \big(\mathbb{E}_x[p(x)]\big)_i \log \bar{p}_i \\
&= H\big(\mathbb{E}_x\big[p(x)\big], \bar{p}\big).
\end{aligned}$$

□

With Lemma 1, we have

$$\begin{aligned}
\log(\text{Inception score}) &= \mathbb{E}_x\big[\mathrm{KL}(C(x) \parallel \bar{C}^G)\big] \\
&= \mathbb{E}_x\big[H\big(C(x), \bar{C}^G\big)\big] - \mathbb{E}_x\big[H\big(C(x)\big)\big] \\
&= H\big(\mathbb{E}_x\big[C(x)\big], \bar{C}^G\big) - \mathbb{E}_x\big[H\big(C(x)\big)\big] \\
&= H(\bar{C}^G) + (-\mathbb{E}_x\big[H\big(C(x)\big)\big]), \\
\log(\text{MODE score}) &= \mathbb{E}_x\big[\mathrm{KL}\big(C(x) \parallel \bar{C}^{\text{train}}\big)\big] - \mathrm{KL}(\bar{C}^G \parallel \bar{C}^{\text{train}}) \\
&= \mathbb{E}_x\big[H\big(C(x), \bar{C}^{\text{train}}\big)\big] - \mathbb{E}_x\big[H\big(C(x)\big)\big] \\
&\quad - H(\bar{C}^G, \bar{C}^{\text{train}}) + H(\bar{C}^G) \\
&= H(\bar{C}^G) + (-\mathbb{E}_x\big[H\big(C(x)\big)\big]), \\
\Rightarrow \text{Inception score} &= \text{MODE score},
\end{aligned} \tag{4}$$

where we see that the required $\bar{C}^{\text{train}}$ is canceled out. Thus, they both consist of two entropy terms: the first term encourages the overall class probability distribution formed by generated samples to be uniformly distributed (large entropy), and the second one encourages the class probability distribution of each generated sample to be sharp (low entropy).

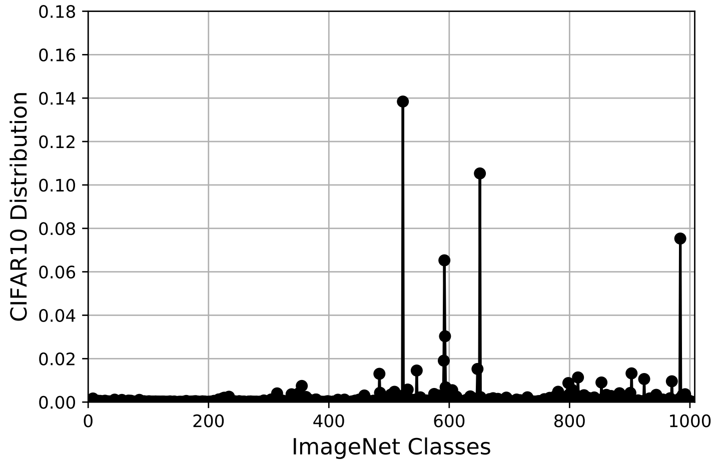

Figure 1: CIFAR-10 training data's overall distribution on ImageNet under the Inception model.

2.2 AM Score

The KL divergence is non-symmetric, and being the reference distribution, $\bar{C}^{\text{train}}$ actually should place at the first place. We here propose to swap $\bar{C}^{\text{train}}$ with its counterpart in the two KL divergence terms Eq. (4), which leads to a more sensible metric:

$$\begin{aligned} &\mathbb{E}_x\left[\mathrm{KL}\left(\bar{C}^{\text{train}} \parallel C(x)\right)\right] - \mathrm{KL}(\bar{C}^{\text{train}} \parallel \bar{C}^G) \\ &= \mathbb{E}_x\left[H\left(\bar{C}^{\text{train}}, C(x)\right)\right] - H(\bar{C}^{\text{train}}) - H(\bar{C}^{\text{train}}, \bar{C}^G) + H(\bar{C}^{\text{train}}) \\ &= \mathbb{E}_x\left[H\left(\bar{C}^{\text{train}}, C(x)\right)\right] + (-H(\bar{C}^{\text{train}}, \bar{C}^G)) \triangleq \text{AM score}. \end{aligned} \tag{5}$$

The above defined AM score is in form of two cross-entropy terms: the first is maximized when each sample is being far away from the training data overall class distribution; the second part is maximized when the generated samples' average distribution is the same as training data. The overall class distribution indicated by the training data, i.e. $\bar{C}^{\text{train}}$, has thus been taken into account. When training data is not evenly distributed, it will be important.

2.3 Pretrained Classifier

It was showed the Inception score with C being the Inception model trained with ImageNet, can well correlate with human evaluation on CIFAR10. We found data of CIFAR10 is not evenly distributed over the ImageNet Inception model, where the entropy term on average distribution of the Inception score does not work well. With a pre-trained CIFAR10 classifier, the AM score can well capture the statistics of average distribution. We hence argue that for general data, the C should be an accordingly pre-trained classifier on given dataset.

Note that the Inception score and the MODE score adopt an exponential transformation based on the above-calculated scores in Eq. (4). With a pre-trained classifier on the given dataset, we will, however, show in the experiment that without the exponential transformation, AM score is informative enough.

2.4 Evaluating AM Score

We have observed that the Inception score and the AM score are fairly consistent with each other when evaluating generative models on CIFAR10, shown in the top of Figure 2.

We show CIFAR10 is not evenly distributed across classes under the Inception model, in figure 1. We further found that, with the Inception model, the entropy terms of the Inception score (Eq. 4) on overall distribution can't work well: as the training goes iteratively, $H(\bar{C}^G)$ keeps oscillating as illustrated in bottom-left of Figure 2. With a pre-trained classifier on CIFGAR10, the AM score (Eq. 5) well captured the statistics on generated samples' overall distribution: $H(\bar{C}^{\text{train}}, \bar{C}^G)$ is stably decreasing, shown in bottom-right of Figure 2.

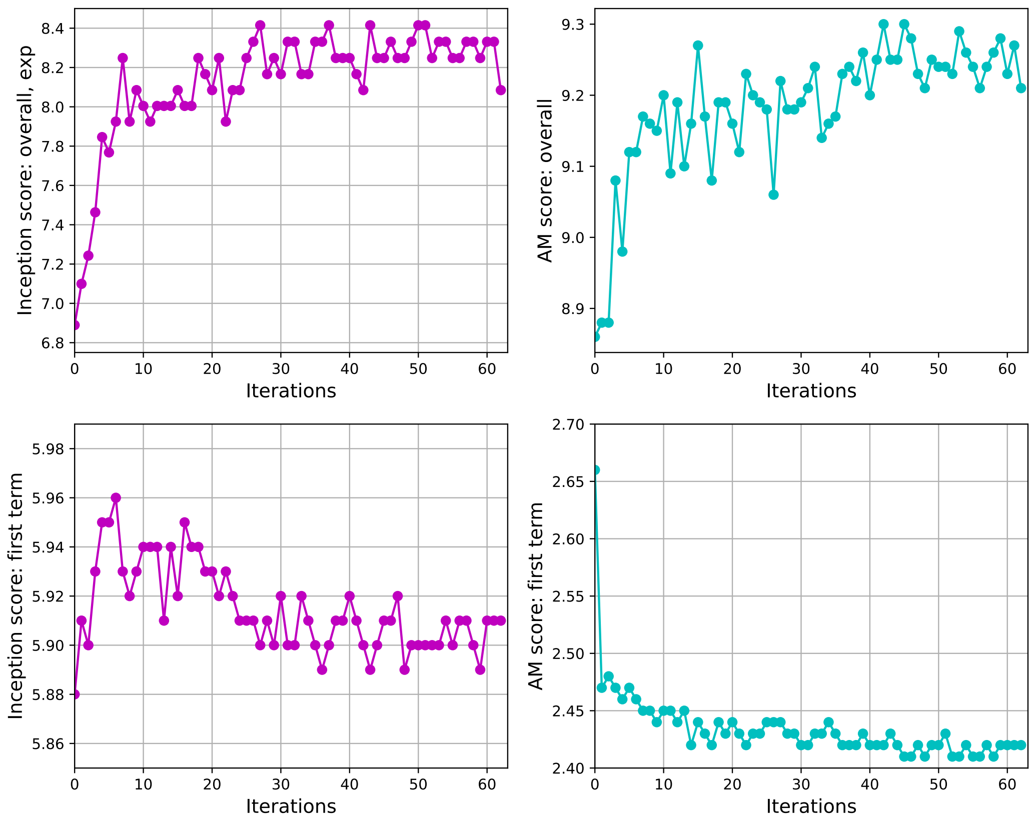

Figure 2: Top: The overall Inception score and AM score. Bottom: The entropy term $H(\bar{D}_c^G)$ of the Inception score on overall distribution. The cross-entropy term $H(\bar{D}_c^{\text{train}}, \bar{D}_c^G)$ of the AM score on overall distribution.

3 Label Smoothing, $-\log(D_r(x))$ Alternative, and Gradient Vanishing

3.1 Label Smoothing

Label smoothing that avoiding extreme logits value was showed to be a good regularization [27]. A general version of label smoothing could be: modifying the target probability of discriminator)

$$\left[\hat{D}_r^D(x), \hat{D}_f^D(x)\right] = \begin{cases} [\lambda_1, 1-\lambda_1] & x \sim G \\ [1-\lambda_2, \lambda_2] & x \sim p_{\text{data}} \end{cases}. \tag{6}$$

Salimans et al. [26] proposed to use only one-side label smoothing. That is, to only apply label smoothing for real samples: $\lambda_1 = 0$ and $\lambda_2 > 0$. The reasoning of one-side label smoothing is applying label smoothing on fake samples will lead to fake mode on data distribution, which is too obscure.

We will next show the exact problems when applying label smoothing to fake samples along with the $\log(1-D_r(x))$ generator loss, in the view of gradient w.r.t. class logit, i.e. the class-aware gradient, and we will also show that the problem does not exist when using the $-\log(D_r(x))$ generator loss.

Lemma 2. *With l being the logits vector and σ being the softmax function, let $\sigma(l)$ be the current softmax probability distribution and $\hat{p}$ denote any target probability distribution, then:*

$$-\frac{\partial H\big(\hat{p}, \sigma(l)\big)}{\partial l} = \hat{p} - \sigma(l). \tag{7}$$

Proof.

$$
\begin{aligned}
&-\left(\frac{\partial H\big(\hat{p},\sigma(l)\big)}{\partial l}\right)_k = -\frac{\partial H\big(\hat{p},\sigma(l)\big)}{\partial l_k} = \frac{\partial \sum_i \hat{p}_i \log \sigma(l)_i}{\partial l_k} = \frac{\partial \sum_i \hat{p}_i \log \frac{\exp(l_i)}{\sum_j \exp(l_j)}}{\partial l_k} \\
&= \frac{\partial \sum_i \hat{p}_i \big(l_i - \log \sum_j \exp(l_j)\big)}{\partial l_k} = \frac{\partial \sum_i \hat{p}_i l_i}{\partial l_k} - \frac{\partial \log\big(\sum_j \exp(l_j)\big)}{\partial l_k} = \hat{p}_k - \frac{\exp(l_k)}{\sum_j \exp(l_j)} \\
&\Rightarrow -\frac{\partial H\big(\hat{p},\sigma(l)\big)}{\partial l} = \hat{p} - \sigma(l).
\end{aligned}
$$

□

3.2 Gradient of Label Smoothing with $\log(1-D_r(x))$

The $\log(1-D_r(x))$ generator loss with label smoothing in terms of cross-entropy is

$$L_G^{\text{log(1-D)}} = -\mathbb{E}_{x\sim G}\Big[H\big([\lambda_1, 1-\lambda_1], [D_r(x), D_f(x)]\big)\Big], \tag{8}$$

the class-aware gradient of which is

$$-\frac{\partial L_G^{\text{log(1-D)}}(x)}{\partial l_r(x)} = D_r(x) - \lambda_1, \tag{9}$$

$$\begin{cases} D_r(x) = \lambda_1 & \text{gradient vanishing} \\ D_r(x) < \lambda_1 & D_r(x) \text{ is optimized towards 0} \\ D_r(x) > \lambda_1 & D_r(x) \text{ is optimized towards 1} \end{cases}. \tag{10}$$

Gradient vanishing is a well know training problem of GAN. Optimizing $D_r(x)$ towards 0 or 1 is also not what desired, because the discriminator is mapping real samples to the distribution with $D_r(x) = 1 - \lambda_2$.

3.3 Gradient of Label Smoothing with $-\log(D_r(x))$

The $-\log(D_r(x))$ generator loss with target $[1-\lambda, \lambda]$ in terms of cross-entropy is

$$L_G^{\text{-log(D)}} = \mathbb{E}_{x\sim G}\Big[H\big([1-\lambda, \lambda], [D_r(x), D_f(x)]\big)\Big], \tag{11}$$

the class-aware gradient of which is

$$-\frac{\partial L_G^{\text{-log(D)}}(x)}{\partial l_r(x)} = (1-\lambda) - D_r(x), \tag{12}$$

$$\begin{cases} D_r(x) = 1-\lambda & \text{stationary point} \\ D_r(x) < 1-\lambda & D_r(x) \text{ towards } 1-\lambda \\ D_r(x) > 1-\lambda & D_r(x) \text{ towards } 1-\lambda \end{cases}. \tag{13}$$

3.4 The $-\log(D_r(x))$ and Gradient Vanishing

Goodfellow et al. [9] proposed to adopt $-\log(D_r(x))$ as an alternative of $\log(1-D_r(x))$ for the generator's loss . The rationale is that, when the discriminator perfectly distinguishes fake samples, the $\log(1-D_r(x))$ loss function of the generator may suffer from the gradient vanishing problem [9, 1]. The motivation of using the $-\log(D_r(x))$ is that: though giving a difference gradient scale, it always preserves the same gradient direction as $\log(1-D_r(x))$.

$\log(1-D_r(x))$ suffers the gradient vanishing problem, and $-\log(D_r(x))$ would avoid the gradient vanishing problem. But we must note that non-zero gradient does not mean that the gradient is efficient or valid. This is a deep topic and detailed discussion is beyond the scope of this paper.

The recent work [1] provides another perspective for the gradient vanishing problem: when the generated distribution is disjoint with the truth data distribution, the gradient from JS divergence is zero. It by default assumed the $\log(1-D_r(x))$, whenever talking about JS divergence for generator's optimization.

The Arjovsky & Bottou [1] is a very good paper, but we suspect the correctness of the Theorem 2.5 about the $-\log(D_r(x))$. The key insight is: the lemma it introduced in the first line of the proof is actually unproved.

One can also view the $-\log(D_r(x))$ as: the generator is doing activation maximization [22, 23, 8] on the log probability of the real class [32]. It is worth mentioning that the maximized activation of one neuron is not necessarily of high quality. Traditionally people introduce various priors [22, 23]. With the existence of fake class, the adversarial process of GAN training ensures the sample quality.

4 Conclusions

We show that Inception score is actually equivalent to Mode score, both consisting of two entropy terms, which would be incompetent when the data is not evenly distributed over classes. We thus propose AM score as an alternative that leverages cross-entropy and takes the prior data distribution into account. Empirical results indicate that AM score outperforms Inception score.

We study label smoothing, gradient vanishing and $-\log(D_r(x))$ alternative from the perspective of class-aware gradient. We show the exact problems when applying label smoothing to fake samples along with the $\log(1-D_r(x))$ generator loss, in the view of class-aware gradient, and more importantly show that the problem does not exist when using the $-\log(D_r(x))$ generator loss.

References

[1] Arjovsky, Martin and Bottou, Léon. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.

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

[3] Berthelot, David, Schumm, Tom, and Metz, Luke. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[4] Brock, Andrew, Lim, Theodore, Ritchie, JM, and Weston, Nick. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[5] Che, Tong, Li, Yanran, Jacob, Athul Paul, Bengio, Yoshua, and Li, Wenjie. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

[6] Dai, Zihang, Almahairi, Amjad, Bachman, Philip, Hovy, Eduard, and Courville, Aaron. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017.

[7] Denton, Emily L, Chintala, Soumith, Fergus, Rob, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pp. 1486–1494, 2015.

[8] Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, and Vincent, Pascal. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341:3, 2009.

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

[10] Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[11] Huang, Xun, Li, Yixuan, Poursaeed, Omid, Hopcroft, John, and Belongie, Serge. Stacked generative adversarial networks. *arXiv preprint arXiv:1612.04357*, 2016.

[12] Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[13] Kim, Taesup and Bengio, Yoshua. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

[14] Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[16] Ledig, Christian, Theis, Lucas, Huszár, Ferenc, Caballero, Jose, Cunningham, Andrew, Acosta, Alejandro, Aitken, Andrew, Tejani, Alykhan, Totz, Johannes, Wang, Zehan, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[17] Li, Chengtao, Alvarez-Melis, David, Xu, Keyulu, Jegelka, Stefanie, and Sra, Suvrit. Distributional adversarial networks. *arXiv preprint arXiv:1706.09549*, 2017.

[18] Lim, Jae Hyun and Ye, Jong Chul. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[19] Mao, Xudong, Li, Qing, Xie, Haoran, Lau, Raymond YK, Wang, Zhen, and Smolley, Stephen Paul. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[20] Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[21] Mroueh, Youssef, Sercu, Tom, and Goel, Vaibhava. Mcgan: Mean and covariance feature matching gan. *arXiv preprint arXiv:1702.08398*, 2017.

[22] Nguyen, Anh, Dosovitskiy, Alexey, Yosinski, Jason, Brox, Thomas, and Clune, Jeff. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pp. 3387–3395, 2016.

[23] Nguyen, Anh, Yosinski, Jason, Bengio, Yoshua, Dosovitskiy, Alexey, and Clune, Jeff. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

[24] Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

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

[26] Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2226–2234, 2016.

[27] Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, and Wojna, Zbigniew. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[28] Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[29] Yu, Lantao, Zhang, Weinan, Wang, Jun, and Yu, Yong. Seqgan: sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv:1609.05473*, 2016.

[30] Zhang, Han, Xu, Tao, Li, Hongsheng, Zhang, Shaoting, Huang, Xiaolei, Wang, Xiaogang, and Metaxas, Dimitris. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016.

[31] Zhao, Junbo, Mathieu, Michael, and LeCun, Yann. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[32] Zhou, Zhiming, Rong, Shu, Cai, Han, Zhang, Weinan, Yu, Yong, and Wang, Jun. Activation maximization generative adversarial nets. *arXiv preprint arXiv:1703.02000*, 2017.