LOGAN-B : Bidirectional Latent Optimized Generative Adversarial Networks

Sharath Ramkumar
University of Massachusetts, Amherst
sharathramku@cs.umass.edu

Abstract

The inductive bias of generative adversarial networks is a powerful tool for learning distributions of data. However, these models are difficult to train and very susceptible to mode collapse, capturing only a few modes of the data distribution. In this work, we introduce bidirectional latent optimized generative adversarial networks as a framework for training generative adversarial networks to mitigate this problem by encouraging the generator to maintain a mapping to the data distribution.

1. Introduction

Generative Adversarial Networks (GANs) are a class of generative models that learn a data distributions. GANs have achieved impressive results on various computer vision tasks such as image generation [10, 4, 11], style transfer [26, 27, 9], and domain adaptation [22, 20, 19] among others. However, GANs are still notoriously difficult to train without careful hyperparameter tuning and susceptible to mode collapse.

1.1. Motivation

The optimal generator in the GAN setup can be seen as an approximation of the training data. In fact, in [7], Goodfellow et al. show that the optimal generator has the same distribution of data samples as the training data. If this is actually true, then it should be possible to sample the training data from the implicit distribution defined by the generator. However, the generator training process is unstable and susceptible to mode collapse, often losing its capability to generate diverse samples. By forcing the generator to maintain a mapping, we hope to encourage the generator to generate diverse samples.

All code will be available at github.com/sharath/logan-b.

1.2. Related Work

1.2.1 LOGAN: Latent Optimization for GANs

Very recently, Wu et al. introduced a similar form of latent optimization in [23] where the latent space is jointly optimized with the generator and discriminator. They use the natural gradient [1] to optimize the latent and show that it results in better IS/FID scores on BigGAN [4]. This definition of latent optimization differs from the one defined by [3], which is the focus of this work.

1.2.2 EBGAN: Energy-based GANs

In [25], Zhao et al. propose EBGAN, a GAN which uses an autoencoder in the discriminator network to encourage the generator network to produce images that contain features that can be used to reconstruct itself. This can be seen as a form of regularization to retain the training data in the generator's implicit distribution.

1.3. Contributions

Our contributions are as follows:

- We introduce latent optimization for an encoder in a bi-directional generative adversarial network using a surrogate reconstruction loss.
- We show that our framework captures the space of the true distribution more accurately than the GAN [7] baseline on a synthetic dataset.
- We show that our framework stabilizes post hoc [6] reconstructions on MNIST and CIFAR10.

2. Background

This section provides a background on GANs, latent optimization, and latent space image embedding.

2.1. Generative Adversarial Networks

GANs were introduced in [7] by Goodfellow et al. In the GAN setting, we wish to find a discriminator D and a generator G that satisfy a mini-max game for a data distribution

$p_{\mathcal{X}}$ and a prior distribution $p_{\mathcal{Z}}$. In practice, we represent G and D using neural networks and we optimize the parameters of the networks in a joint manner alternating between each network, in a manner that can be seen as two players taking turns:

$$\min_G \max_D \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\log(D(x))] + \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\log(1 - D(G(z)))] . \tag{1}$$

2.2. Generative Latent Optimization

The generative latent optimization (GLO) framework was proposed by Bojanowski et al. in [3]. For a large set of images, $\{x_i\}_{i=1}^N$, with each image $x_i \in \mathcal{X}$, we assign a d-dimensional vector $z_i \in \mathcal{Z}$ to each image. Then perform a joint optimization of the parameters of the generator network and the noise vector for the input to the network:

$$\min_G \frac{1}{N} \sum_{i=1}^{N} \left[\min_{z_i \in \mathcal{Z}} \ell(G(z_i), x_i) \right] . \tag{2}$$

2.3. Latent Space Embedding

A variety of methods have been proposed to find latent embeddings for natural images in the Generator's latent space. In this work, we compare against the Bidirectional GAN (BiGAN) [5] and variants of optimization [2] on the input. More formally, the latent space embedding z of a natural image x can be written as the following optimization problem:

$$\min_{z \in \mathcal{Z}} \ell(G(x), x). \tag{3}$$

In (3), ℓ is a distance metric between two images. Possible choices for ℓ include L1/L2, perceptual distance [24], and Laplacian pyramid loss [14] among others.

2.3.1 Input Optimization

One approach to obtaining an embedding for an image is by optimizing the input directly. Other variations of this include *stochastic clipping* for priors with finite supports, proposed by Tripathi et al. in [15] and layer-wise optimization proposed by Bau et al. in [2]. In this approach, we initialize an input noise vector by sampling from the prior and then perform gradient descent on the input to minimize the distance metric.

2.3.2 Bidirectional Generative Adversarial Networks (BiGANs)

Another approach to obtain latent space image embeddings is to pass the natural image through an encoder network that is trained jointly with the the Generator. This jointly trained network is called an *ad hoc encoder*. In the BiGAN framework, the generator and encoder are inverses of each other at the global optima [5, 6]. The optimization problem is framed as the following:

$$\min_{G,E} \max_D \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\log(D(x, E(x)))] + \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\log(1 - D(G(z), z))] . \tag{4}$$

2.4. Performance Metrics

This section provides an overview of the metrics we use to measure performance.

2.4.1 Inception Score

Inception score (IS) is a metric for evaluating the quality of samples from a generator [18]. The score is calculated using features from the Inception network [21]. Salimans et al. found that that the IS metric is correlated with human evaluation. Higher values of IS correspond to a "better" generator network.

2.4.2 Fréchet Inception Distance

Fréchet Inception Distance (FID) is a distance metric proposed by Heusel et al. in [8] as an improvement over the IS metric by using the statistics of the training dataset. The FID metric has also been shown to be more robust to disturbances to samples such as noise and occlusions. This is a distance metric, so lower values of FID correspond to a "better" generator network.

2.4.3 Reconstruction Error

While IS and FID are reasonable [17] metrics for sample quality, we also report the average reconstruction error of an encoder trained after the generator. We refer to this as the *post hoc* [6] encoder, in contrast to the BiGAN encoder which is *ad hoc*. We train an encoder network using the latent random noise as the target and images from the generator as the input 5 times. Then we compute the mean and standard deviation of the reconstruction losses on the hold-out data and report that. Low values for reconstruction error indicate:

1. the extent to which the generator captures the distribution of the training data

2. and the invertibility of the generator, which can be seen as a measure of latent optimality.

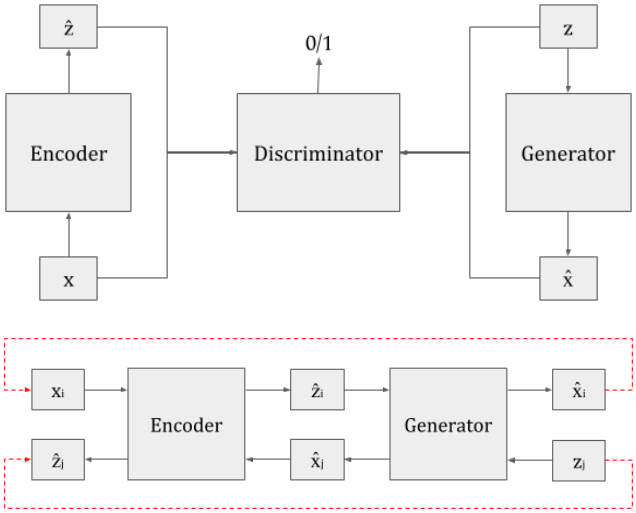

Figure 1. Overview of the adversarial and surrogate setup for LOGAN-B. First we update discriminator, generator, and encoder using adversarial losses. Then we update the encoder and generator using the surrogate reconstruction loss.

3. Bidirectional Latent Optimized Generative Adversarial Network

The bidirectional latent optimized GAN (LOGAN-B) jointly trains an encoder E, generator G, and discriminator D with the following loss functions:

$$\mathcal{L}_G = \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\log(1 - D(G(z), z))] + \mathcal{L}_S \tag{5}$$

$$\mathcal{L}_E = \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\log(D(x, E(x)))] + \mathcal{L}_S \tag{6}$$

$$\mathcal{L}_D = - \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\log(D(x, E(x)))] - \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\log(1 - D(G(z), z))] \tag{7}$$

where $\mathcal{L}_S$ is the surrogate reconstruction loss:

$$\mathcal{L}_S = \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(x, G(E(x)))] \tag{8}$$

3.1. Relationship to BiGAN and GLO

In the BiGAN framework, the optimal encoder is capable of inverting the optimal generator [5]. This is a form of latent optimization as defined by Bojanowski et al. in [3]:

$$\min_G \frac{1}{N} \sum_{i=1}^{N} \left[\min_{z_i \in \mathcal{Z}} \ell(x_i, G(z_i)) \right] \tag{9}$$

$$\min_{G,E} \frac{1}{N} \sum_{i=1}^{N} [\ell(x_i, G(E(x_i)))] \tag{10}$$

$$\approx \min_{G,E} \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(x, G(E(x)))] \tag{11}$$

The problems defined by (10) and (11) are equal as N approaches ∞ [5] . Under the same conditions, the optimal generator and encoder in the BiGAN framework also satisfy all of these optimization problems:

$$\min_{G,E} \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(x, G(E(x)))] \tag{12}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(E(x), E(G(E(x))))] \tag{13}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(x, G(E(G(E(x)))))] \tag{14}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{x \sim p_{\mathcal{X}}} [\ell(E(x), E(G(E(G(E(x))))))] \tag{15}$$

...

Under the same conditions, these are also all equivalent optimization problems over the latent space:

$$\min_{G,E} \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\ell(z, E(G(z)))] \tag{16}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\ell(G(z), G(E(G(z))))] \tag{17}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\ell(z, E(G(E(G(z)))))] \tag{18}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\ell(G(z), G(E(G(E(G(z))))))] \tag{19}$$

$$\min_{G,E} \mathop{\mathbb{E}}_{z \sim p_{\mathcal{Z}}} [\ell(z, E(G(E(G(E(G(z)))))))] \tag{20}$$

...

Donahue et al. propose (16) in [5] as the *latent regressor*. They show that using (16) performs poorly since the generator is not actually optimal during training. Furthermore, they argue that this has limited benefits: Since the encoder never sees the training data, the loss biases the optimization towards a local optima near its current parameters rather than near the global minimum. To rectify this, we propose using samples from the training data to minimize (12) as the *surrogate loss*. We speculate that other combinations may contain useful gradients, but leave them to future work.

4. Experiments

4.1. Synthetic Data

We consider a synthetic dataset of 8 Gaussian distributions and report the reconstruction error (as defined in 2.4.3) in Table 1. We use the L1 distance function for ℓ and a uniform noise distribution for the prior. Post hoc encoders are initialized randomly and trained to estimate the input to the generator that produced the samples. The ad hoc encoder

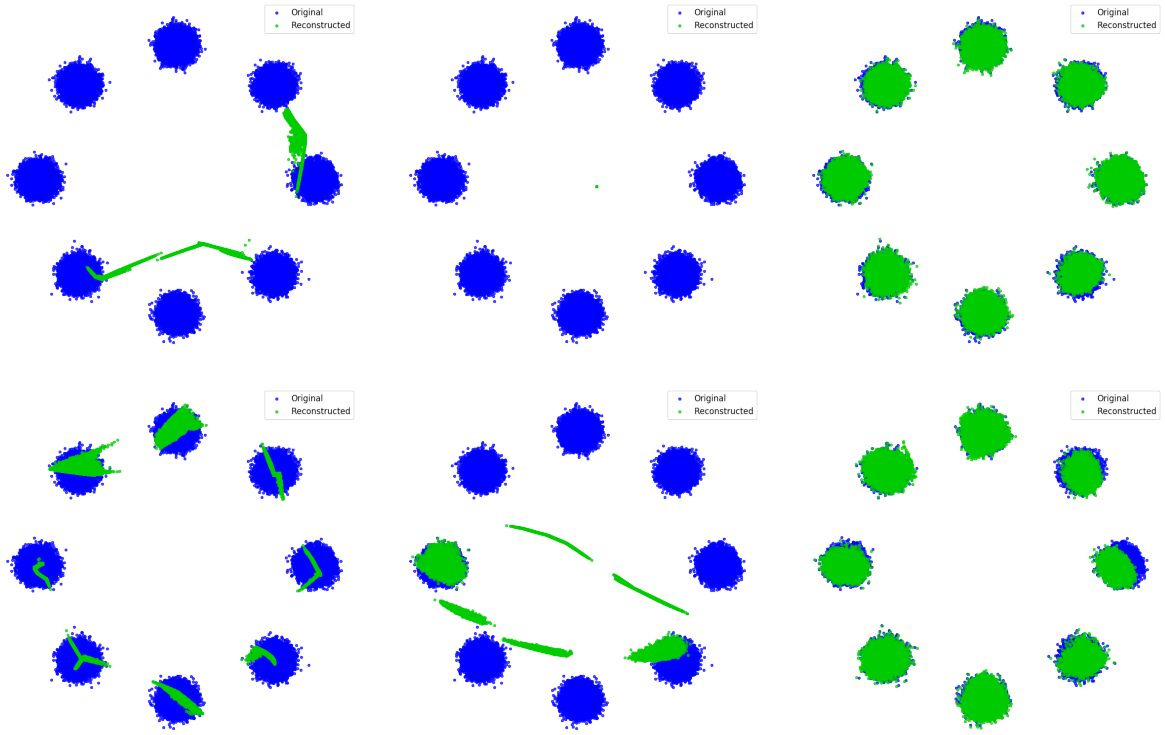

Figure 2. Reconstructions on the synthetic Gaussian mixture dataset. Reconstructions for GAN, BiGAN, and LOGAN-B (left-to-right) for both ad hoc (top) and post hoc (bottom) trained encoders. The BiGAN ad hoc encoder collapsed, but the post hoc encoder shows a more accurate view of the performance of the BiGAN generator.

Method	ad hoc	post hoc
GAN	2.7937	0.1261 ± 0.0060
BiGAN	2.5036	1.4310 ± 0.0744
LOGAN-B	0.0271	0.0474 ± 0.0057

Table 1. Ad hoc/post hoc reconstruction errors on the synthetic dataset.

for the GAN does not affect training, it is simply updated with the generator.

The reconstructions (Figure 4.1) that the BiGAN encoder collapses to during training always map to the center of the ring. However, the post hoc encoder for the same generator network shows that the generator still maintains mappings from the latent space to some of the Gaussian distributions in the training data.

4.2. MNIST

For our experiments on the MNIST [13] dataset, we use LOGAN-B with the DCGAN [16] architectures for the generator and discriminator and an inverted DCGAN generator for the encoder. MNIST images were resized to 32×32 and normalized. Performance metrics were computed after the 10th epoch.

Method	ad hoc	post hoc	IS	FID
DCGAN	290.82	101.76 ± 3.76	2.13	71.68
BiGAN	720.15	720.04 ± 0.49	1.00	431.01
LOGAN-B	29.99	31.01 ± 1.33	2.01	40.23

Table 2. Ad hoc/post hoc reconstruction errors and IS/FID scores for MNIST. Lower FID is better. Higher IS is better.

4.3. CIFAR10

For our experiments on the CIFAR10 [12] dataset, we use LOGAN-B with the DCGAN [16] architectures for the generator and discriminator and a inverted generator for the encoder. CIFAR10 images were normalized. Performance metrics were computed after the 25th epoch.

Method	ad hoc	post hoc	IS	FID
DCGAN	1649.95	919.49 ± 11.34	2.42	227.70
BiGAN	1567.17	1578.30 ± 1.54	1.00	426.72
LOGAN-B	590.09	372.04 ± 4.38	4.04	140.63

Table 3. Ad hoc/post hoc reconstruction errors and IS/FID scores for CIFAR10. Lower FID is better. Higher IS is better.

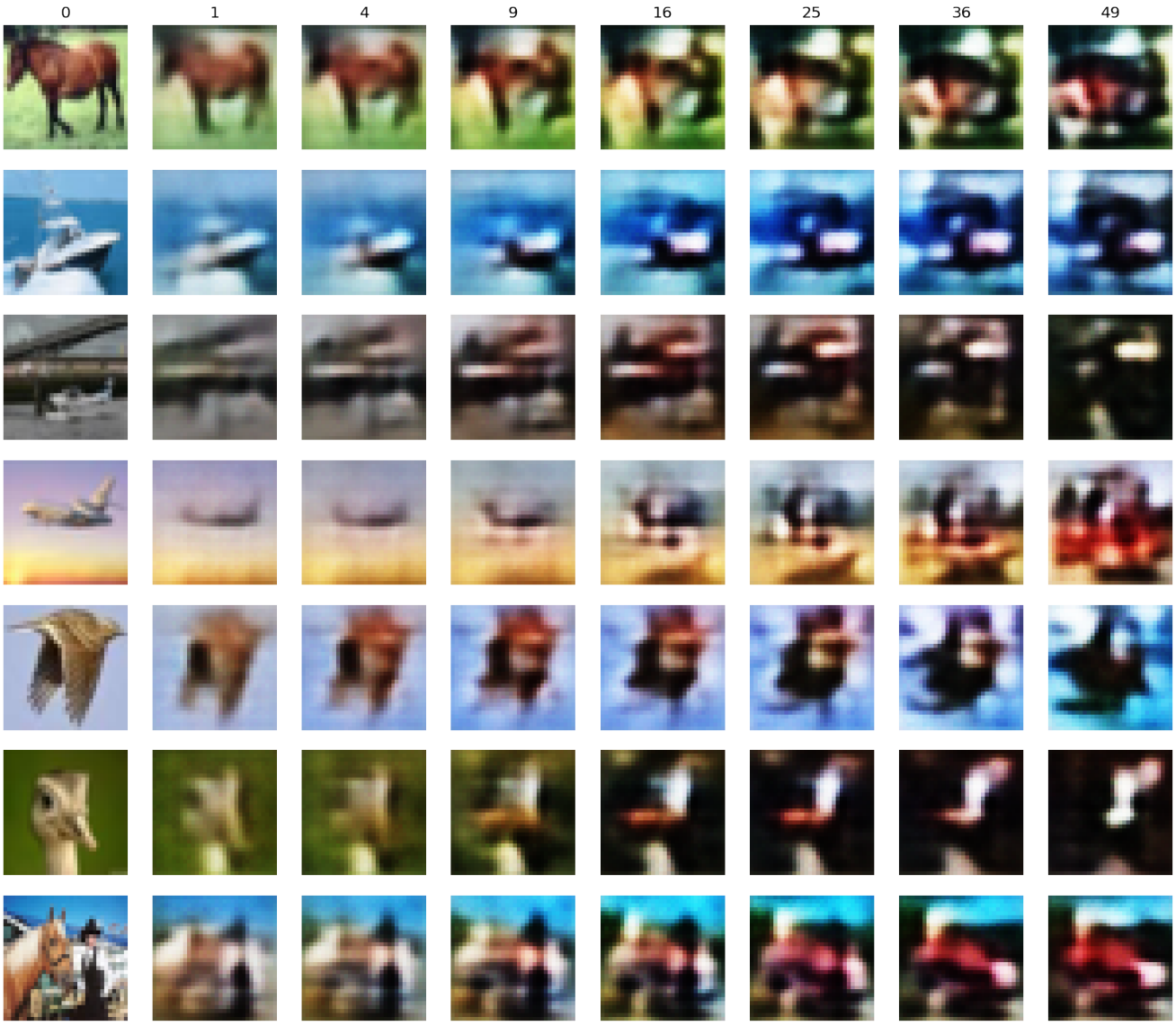

Figure 3. Bootstrapping results on the CIFAR10 dataset. The first column contains the original images from the dataset. The number of bootstraps increases from left-to-right and is indicated at the top.

4.4. Bootstrapping

One way to qualitatively visualize the reconstruction loss is by *bootstrapping*: sampling from the training data, passing through the encoder and then passing the latent embedding again through the generator. The results from performing this on CIFAR10 are shown in Figure 3. Interestingly, the last few columns seem to not change despite the number of bootstraps growing exponentially.

5. Discussion

The results indicate that the BiGAN training is the least stable of the three models that we considered. This instability is most likely caused by a lack of proper hyperparameters. In practice, it can be expensive to tune hyperparameters when the space of hyperparameters that work is small. This is an additional weakness of the BiGAN model. We have introduced a form of latent optimization for a bidirectional GAN (LOGAN-B) and shown improved ability to capture the space of the true distribution. We have also shown that the generator in the LOGAN-B framework can be more easily inverted due to the latent optimization provided by real training data. A limitation to this work is that we only considered L1 loss for reconstruction. This type of loss is unlikely to scale for larger natural image datasets. Future work in this area should keep that in mind.

References

[1] Shun-Ichi Amari. Natural gradient works efficiently in learning. Neural computation, 10(2):251–276, 1998.

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In Proceedings of the IEEE International Conference on Computer Vision, pages 4502–4511, 2019.

[3] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776, 2017.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016.

[6] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

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

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems, pages 6626–6637, 2017.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[10] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. arXiv preprint arXiv:1912.04958, 2019.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[14] Haibin Ling and Kazunori Okada. Diffusion distance for histogram comparison. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 1, pages 246–253. IEEE, 2006.

[15] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. arXiv preprint arXiv:1702.04782, 2017.

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

[17] Suman Ravuri and Oriol Vinyals. Seeing is not necessarily believing: Limitations of biggans for data augmentation. 2019.

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

[19] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8503–8512, 2018.

[20] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1–4, 2019.

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

[22] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7167–7176, 2017.

[23] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. arXiv preprint arXiv:1912.00953, 2019.

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 586–595, 2018.

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

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232, 2017.

[27] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In Advances in Neural Information Processing Systems, 2017.