Improved Training of Generative Adversarial Networks Using Representative Features

Duhyeon Bang \(^1\) Hyunjung Shim \(^1\)

Abstract

Despite of the success of Generative Adversarial Networks (GANs) for image generation tasks, the trade-off between image diversity and visual quality are an well-known issue. Conventional techniques achieve either visual quality or image diversity; the improvement in one side is often the result of sacrificing the degradation in the other side. In this paper, we aim to achieve both simultaneously by improving the stability of training GANs. A key idea of the proposed approach is to implicitly regularizing the discriminator using a representative feature. For that, this representative feature is extracted from the data distribution, and then transferred to the discriminator for enforcing slow updates of the gradient. Consequently, the entire training process is stabilized because the learning curve of discriminator varies slowly. Based on extensive evaluation, we demonstrate that our approach improves the visual quality and diversity of state-of-the-art GANs.

1. Introduction

Generative models aim to solve the problem of density estimation by learning the model distribution \(P_{\text{model}}\), which approximates the true but unknown data distribution of \(P_{\text{data}}\) using a set of training examples drawn from \(P_{\text{data}}\) (Goodfellow, 2016). Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a family of generative models capable of implicitly estimating a data distribution without an analytic formula or variational bound of \(P_{\text{model}}\). Existing GANs have been mainly used for image generation tasks, with which they have showed impressive results and produced sharp and realistic images of natural scenes. Because of the flexible nature of the model definition and high quality results, GANs have been applied to many real-world applications, including super-resolution, colorization, face generation, image completion, etc. (Bao et al., 2017; Ledig et al., 2016; Yeh et al., 2017; Cao et al., 2017)

Training a GAN involves two separate networks with competitive goals: a discriminator, \(D\), to distinguish between the real and fake data and a generator, \(G\), to create data to be as real as possible to fool the discriminator. Consequently, the generator implicitly models \(P_{\text{model}}\) which approximates \(P_{\text{data}}\). In (Goodfellow et al., 2014), this problem is formulated by the following minimax game as

\[
\min_{G} \max_{D} \mathbb{E}_{x \sim P_{\text{data}}} \left[ \log(D(x)) \right] + \mathbb{E}_{z \sim P_{z}} \left[ \log(1 - D(G(z))) \right].
\]

When the generator produces perfect samples (meaning that \(P_{\text{model}}\) is identical to \(P_{\text{data}}\)), the discriminator cannot distinguish between the real and fake data. Subsequently, this game ends because it reaches a Nash equilibrium.

Although GANs have been successful in the field of image generation, the instability of training processes such as extreme sensitivity of a network structure and parameter tuning are well-known disadvantages. Training instability has yielded two major problems, namely gradient vanishing and mode collapse. As discussed in (Arjovsky & Bottou, 2017), gradient vanishing becomes a serious problem when any subset of \(P_{\text{data}}\) and \(P_{\text{model}}\) are disjointed such that the discriminator separates real and fake data perfectly; i.e. the generator no longer improves the data because the discriminator has reached its optimum. This yields poor results because training stops even though \(P_{\text{model}}\) has not learned \(P_{\text{data}}\) properly. Mode collapse is another common problem of GANs; the generator repeatedly produces the same or similar output because \(P_{\text{model}}\) only encapsulates the major or single modes of \(P_{\text{data}}\) in order to easily fool the discriminator.

Unfortunately, the trade-off between image quality and mode collapse are theoretically and empirically investigated in previous work (Berthelot et al., 2017; Fedus et al., 2017). Existing studies often achieved either visual quality or image diversity, but not both. Their results can be interpreted as the visual quality can be achieved by minimizing the reverse KL divergence while the image diversity is strongly...
correlated with minimizing the forward KL divergence. To break the trade-off, recent techniques (Kodali et al., 2017; Gulrajani et al., 2017; Fedus et al., 2017) introduce the gradient penalty for regularizing the divergence (or distance) for training GANs. The gradient penalty smooths out the learning curve, thus improving the stability of training. As a result of stabilized training, the gradient penalty is effective to improve both visual quality and image diversity, and this idea is evaluated in various architectures of GANs.

The goal of the proposed approach is similar to the gradient penalty in that we also aim to stabilize the training in order to break the trade-off between the visual quality and image diversity. To this end, we propose an unsupervised approach to implicitly regularizing the discriminator using representative features. Unlike the gradient penalty term, our approach does not modify the objective function of GANs (i.e., use the same divergence or loss definition as a baseline GAN). Instead, we introduce the representative features from a pre-trained autoencoder (AE) and transfer them to a discriminator for training GANs. Representative features are a good representation for reconstructing the overall data distribution $P_{data}$, but not as distinguishable as the discriminative features (i.e., features learnt for classification). Thereby, the discriminator is implicitly interrupted by representative features for the discrimination while encouraged to account for the entire data distribution.

It is worthwhile noting that this pre-trained AE learns from samples of $P_{data}$ and is then fixed. By isolating the representative feature extraction from training GANs, it is possible to guarantee that the embedding space of pre-trained AE and corresponding features exhibit the representative power. Because the representative features are derived from the pre-trained network, they are much informative at the early stage of training discriminator; accelerating the early stage training of GANs. However, from the second half of training, the representative features no longer distinguish real and fake images. At this stage, the overall loss of our model consists of the loss of representative features and that of discriminative features. Because the loss of representative feature is nearly constant at the second half of training, the overall loss of discriminator varies slowly over iterations. As a result, we stabilize GAN training, thereby it is effective to improve both the visual quality and image diversity of generated samples. We name this new architecture as a Representative Feature based on a Generative Adversarial Network (RFGAN).

The major contributions of this paper are as follows:

1. We employ additional representative features extracted from a pre-trained AE for implicitly constraining the discriminator’s update. In this way, the learning curve of GANs varies slowly, thus the GAN training is stabilized. As a result, we simultaneously achieve the visual quality and image diversity of GANs in an unsupervised manner.

2. Our framework of combining the pre-trained AE is easily extendable to various GANs using different divergences or structures. Also, the proposed model is robust against the parameter selections; all results in this paper use the same hyper-parameters suggested by a baseline GAN.

3. We conduct extensive experimental evaluations to show the effectiveness of RFGAN; our approach improves existing GANs including GANs with the gradient penalty.

In Section 2, we review recent studies and analyze the relationships of RFGAN with them. Next, the architecture and distinctive characteristics of RFGAN are demonstrated in Section 3. In Section 4, we summarize the results of extensive experiments using both simulated and real data. Based on both quantitative and qualitative evaluations, we show that the RFGAN improved image quality and also achieved diversity in data generation.

2. Related Work

A variety of techniques have been proposed in the past for improving the stability of GAN training which mostly aim to resolve gradient vanishing and mode collapse. For that, existing studies can be categorized into two groups.

A. GAN training by modifying the network design

To avoid gradient vanishing, (Goodfellow et al., 2014) proposed modifying a minimax game-based GAN formulation into a non-saturating game. Consequently, the objective function of the generator was changed from $J(G) = \mathbb{E}_{z \sim P_z} \log(1 - D(G(z)))$ to $J(G) = -\frac{1}{2}\mathbb{E}_{z \sim P_z} \log(D(G(z)))$. This simple modification effectively resolved the problem of gradient vanishing, and several studies (Arjovsky & Bottou, 2017; Fedus et al., 2017) have confirmed the same conclusion based on theoretical and empirical analysis. (Radford et al., 2015) first introduced GAN with a stable deep convolutional architecture (DCGAN), and their visual quality was quantitatively superior to a variant of GANs proposed later, according to (Lucic et al., 2017).

Unfortunately, mode collapse was a major weakness of DCGAN. To tackle this problem, unrolled GAN was proposed by (Metz et al., 2016) to adjust the gradient update of the generator by introducing a surrogate objective function, which simulated the response of discriminator upon the changes of the generator. As a result, unrolled GAN successfully solved model collapse. InfoGAN (Chen et al., 2016) achieved an unsupervised disentangled representation by minimizing the mutual information of auxiliary loss (i.e., matching the semantic information) and adversarial loss. Additionally, (Salimans et al., 2016) proposed various ways to stabilize the training process of GAN using
semi-supervised learning and smoothing labeling.

**B. Effects of various divergences**

In (Nowozin et al., 2016), the authors showed that the Jensen-Shannon divergence used in the original GAN formulation (Goodfellow et al., 2014) can be extended to different divergences, including f-divergence (f-GAN). (Arjovsky & Bottou, 2017) and (Arjovsky et al., 2017) analyzed a source of instability when training GAN and theoretically showed that the Kullback–Leibler (KL) divergence was one of the causes. Later, they proposed the Wasserstein distance to measure the similarity between $P_{\text{model}}$ and $P_{\text{data}}$ in order to overcome the instability when training GAN. In order to implement the Wasserstein distance to produce Wasserstein GAN, (Arjovsky et al., 2017) introduced the k-Lipschitz constraint into the discriminator, which is enforced by weight clipping. However, (Gulrajani et al., 2017) pointed out that weight clipping often fails to capture the higher moments of $P_{\text{data}}$, and the authors suggested a gradient penalty for better modeling of $P_{\text{data}}$. (Kodali et al., 2017) showed that the discriminator became closer to the convex set by using the gradient penalty as a regularization term, and this is effective in improving the stability of training GAN. For Least Squares GAN (LSGAN), (Mao et al., 2017) replaced the Jensen-Shannon divergence defined by the sigmoid cross-entropy loss term to the least squares loss term, and showed that GAN with least squares loss is essentially interpreted as the minimization of the Pearson $\chi^2$ divergence.

To summarize, most existing approaches for GAN have suggested finding stable architecture or additional layers to stabilize the discriminator’s updates, to change the divergence, or to add a regularization term to stabilize the discriminator. The proposed algorithm, RFGAN, can be classified into the first category, which modifies the architecture of GAN. RFGAN is distinguishable from other GANs in that features from the encoder layers of the pre-trained AE are transferred while training the discriminator. Several existing approaches also use AE or encoder architectures, and they are explained as follows.

ALI(Dumoulin et al., 2016), BiGAN(Donahue et al., 2016) and MDGAN(Chen et al., 2016) suggested that the samples from $P_{\text{data}}$ should be mapped to the latent space of the generator using the encoder structure. This is to enforce the latent space of the generator to learn the entire distribution of $P_{\text{data}}$, thus solving mode collapse. Although these are similar approaches to ours in that the encoder layers are employed to develop GAN, our RFGAN uses an AE for a completely different purpose by an alternative method to extract representative features, and furthermore, those features are to stabilize the discriminator. EBGAN(Zhao et al., 2016) and BEGAN(Berthelot et al., 2017) suggested an energy-based function to develop the discriminator, by which their networks showed stable convergence and were less sensitive to parameter selection. In both studies, they borrowed the AE architecture for defining the energy-based function, which served the discriminator. Our model employs representative features from the encoder layers and still includes the conventional discriminator architecture to maintain the discriminative power of the discriminator. (Warde-Farley & Bengio, 2016) extracted the features from the discriminator layers, applied a denoising AE, and used its output to regularize the adversarial loss. Based on this denoising feature matching, they improved the quality of image generation and employed the denoising AE to ensure the robust features of the discriminator. The proposed model is different in that the AE is used as the feature extractor.

Additionally, unlike all of the existing approaches that have trained the AE or encoder layers as part of the GAN architecture, the proposed algorithm separately trains the AE to learn $P_{\text{data}}$ in an unsupervised manner. In this way, we disconnect the feedback from $P_{\text{model}}$ when training the AE, and so focus on learning the feature space for representing $P_{\text{data}}$. Moreover, we do not utilize the decoder, thus preventing problems such as image blur.

### 3. The RFGAN model

To resolve the instability of training GAN, we extract the representative features from an pre-trained AE and transfer them to the discriminator; this implicitly enforces that the discriminator is slowly updated once the generator produces beyond the certain level of visual quality.

The aim of an AE is to learn a reduced representation of the given data because it is formulated by reconstructing the input data after passing through the network. Consequently, feature spaces learnt by the AE are powerful representations for reconstructing the $P_{\text{data}}$ distribution.

By focusing on the functionality of the AE as a feature extractor, several studies have utilized it for classification tasks through fine-tuning (Zhou et al., 2012; Chen et al., 2015). On the contrary, the argument by (Alain & Bengio, 2014; Wei et al., 2015) noted that good representation for reconstruction does not guarantee good classification. This is a reasonable statement because the features of the reconstruction model and those of the discriminative model are derived from different objectives, thus should be applied to the appropriate tasks for optimal performance.

When training a GAN, the discriminator operates as a binary classifier (Radford et al., 2015), so the features extracted from the discriminator specialize in distinguishing whether the input is real or fake. This means that the discriminator’s features are totally different properties from the AE’s features. By focusing on these different properties, we denote the AE’s features and those of the discriminator as representative and discriminative features, respectively. While
the original formulation of GAN evaluates the quality of data generation purely based on discriminative features, we propose leveraging both representative and discriminative features to implicitly regularize the discriminator in order to stabilize GAN training.

In this section, we describe the proposed model, RFGAN, and the effect of the modified architecture for training the discriminator. Furthermore, we investigate how this effect could overcome mode collapse and improve visual quality.

3.1. The architecture of RFGAN

The main contribution of our model is in adopting representative features from pre-trained AE to develop GAN, thus our model is built based on various GAN architectures. Hence, RFGAN refers to a set of GANs using the representative features. For simplicity, we use DCGAN (Radford et al., 2015) employing non-saturated loss as a baseline GAN, and apply representative features to discriminator to construct DCGAN-RF. In Section 4, we use various other GANs as baselines to develop RFGAN variants. Throughout this paper, we use exactly the same hyper-parameters, metrics, and settings as suggested for a baseline GAN, to show that our model is indeed insensitive to parameter selection. Instead, we supply representative features extracted from the encoder layer (part of pre-trained AE) to the discriminator. Note that the AE is pre-trained in an unsupervised manner using samples from $P_{data}$ and isolated from the GAN training.

More specifically, we construct the AE in a way that its encoder and decoder share the same architecture as the discriminator and generator, respectively. Next, we concatenate two feature vectors, one extracted from the last convolution layer of the encoder and the other from the discriminator. Given the concatenated feature vector, final weights are trained for deciding between real or fake input. This process is demonstrated by the simplified graphical model shown in Figure 1, in which the input data passes through two networks, encoder $E$ and discriminator $D$. In this figure, $h_1$ and $h_2$ represent the representative and discriminative feature vectors, respectively. These two feature vectors are concatenated and transformed to a single sigmoid output $Y$ through a fully connected layer. The output is evaluated with the ground truth label based on sigmoid cross entropy, and then the gradient of the loss function is delivered to the discriminator via backpropagation to update the parameters. Note that this feedback is not propagated to the encoder because its parameters are already trained and then fixed.

The procedure for gradient updates is defined as follows:

$$D(x) = - \log Y \text{ for } x \sim P_{data}, \ Y = \sigma(h_1w_1 + h_2w_2),$$

$$\nabla_{w_1} = \frac{\partial D(x)}{\partial w_1} = - \frac{1}{Y} \cdot (1 - Y) \cdot h_i = (Y - 1) \cdot h_i,$$

$$\nabla_{w_D} = \frac{\partial D(x)}{\partial w_D} = (Y - 1) \cdot w_2 \cdot u(w_D).$$

The GAN objective function represented by parameters

$$J_0(\theta_G, \theta_D) = \mathbb{E}_{x \sim P_{data}}[\log D(x; \theta_D)] + \mathbb{E}_{z \sim \mathcal{N}}[\log (1 - D(G(z; \theta_G); \theta_D))],$$

which is updated by

$$\theta_D^{t+1} \leftarrow \theta_D^{t} - \eta_D \frac{df(\theta_G, \theta_D^{t})}{d\theta_D} = \theta_D^{t} - \eta_D (\nabla_{w_D} + \nabla_{w_1}),$$

$$\theta_G^{t+1} \leftarrow \theta_G^{t} - \eta_G \frac{df(\theta_G, \theta_D^{t})}{d\theta_G^{t}},$$

where $\sigma$ and $u$ are a sigmoid and step function, respectively.

Since the encoder is pre-determined, we only consider the update of the discriminator. After this, we derive the gradient toward the discriminator by calculating the partial derivative of loss term with respect to $w_D$, which indicates the network parameters as shown in Figure 1. From this formulation, we observe that $\nabla w_D$ depends on $h_1$. In this way, the representative features affect the discriminator update. This procedure is derived by the case where $x$ is real. In the case of a fake sample, the same conclusion is deduced except that $D(x)$ is now changed to $-\log(1 - \sigma(h_1w_1 + h_2w_2))$. Based on the aforementioned formulation, the generator is trained by considering both the representative and discriminative features because it should fool the discriminator by maximizing $-\log D(G(z))$. Again, it is notable that our representative features keep their own properties such as a global representation for reconstructing the data distribution by fixing the encoder parameters.

3.2. Mode collapse

Based on probabilistic interpretation, the decoder of the AE estimates the parameters of a distribution $P(x|En(x))$ that may generate $x$ with high probability formulated by cross-entropy loss (Vincent et al., 2010). We could deem
that the AE follows the forward KL divergence between $P_{data}$ and $P_{model}$ (i.e. $\text{KL}(P_{data} \parallel P_{model})$). Since the model approximated by the forward KL divergence is evaluated using every true data sample (i.e. any $x$ for $P_{data}(x) > 0$), it tends to average out all modes of $P_{data}$ (Goodfellow, 2016). Hence, we can expect that the representative features extracted from the AE are similar in that they are effective at representing entire modes of $P_{data}$.

On the contrary, the aim of DCGAN with a non-saturated loss (the base architecture of our model) is to optimize the reverse KL divergence between $P_{data}$ and $P_{model}$: more specifically, $\text{KL}(P_{model} \parallel P_{data}) - 2\text{JSD}$ (Arjovsky & Bottou, 2017), where JSD is the notion of the Jensen–Shannon divergence. Because the model based on the reverse KL objective is examined for every fake sample (i.e. any $x$ where $P_{model}(x) > 0$), it has no penalty for covering the entire true data distribution; hence, it is likely to focus on the single or partial modes of the true data distribution. This phenomenon is a well-known mode collapse problem.

Our RFGAN performs the optimization of the reverse KL divergence because our framework is built upon non-saturated GAN. At the same time, we introduce representative features from the AE, which encourages the model to cover the entire modes of $P_{data}$, like optimizing the forward KL divergence. This leads to suppressing the tendency toward falling mode collapse.

### 3.3. Improving visual quality

While the representative features are useful to advance the discriminator in the early stage, they become less informative when approaching the second half of training. This behavior is expected because the AE shows limited performance during the discrimination task. Note that the AE is built upon minimizing the reconstruction error (e.g. L2 or L1 loss); this model cannot learn multiple different, correct answers, which causes the model to choose the averaged (or median) output (Goodfellow, 2016). This property is still useful to distinguish between poor fake and real input. However, when the generator starts producing good fake input, the representative features from the AE are less discriminative, thus rather interfere with decisions by the discriminator in the later stages of training. Figure 2 shows the output when several real and fake examples passed through the pre-trained AE. At the beginning of training, it is easy to distinguish whether the input is real or fake. Yet, after several iterations, we observed that both look quite similar. These experimental results demonstrate the discriminative power of the AE for different levels of fake examples.

Consequently, it is difficult to improve the visual quality of data generation beyond a certain level using representative features alone. This is why the proposed model employs both representative and discriminative features to train the discriminator. Although the representative features interfere with discrimination between the real and fake input after the training goes on, the discriminator using the proposed model still has discriminative features, which allows the training to continue. As a result, the generator consistently receives sufficient feedback (i.e. the gradient from the discriminator is increased.) from the discriminator to learn $P_{data}$. Furthermore, because these two features are two opposing forces and disagree with each other, in the end the discriminator slowly approaches saturation. By slowing down the growth of the discriminator, we observe that our model generates high quality data, thereby improving the existing GAN.

### 4. Experimental results

For both the quantitative and qualitative evaluations, we conducted experiments using simulated and three real datasets; CelebA (Liu et al., 2015), LSUN-bedroom (Yu et al., 2015), and CIFAR-10 (Krizhevsky & Hinton, 2009) by normalizing between -1 and 1. During the experiment, denoising AE (Vincent et al., 2008) is used to improve the robustness of the feature extraction and slight quality improvement is achieved compared to the conventional AE.

#### 4.1. Handling mode collapse

To evaluate how well the proposed model could achieve the diversity of data generation (i.e. solving mode collapse), we trained our network with a simple 2D mixture of 8 Gaussians suggested by (Metz et al., 2016). The mean of each Gaussian formed a ring and each Gaussian distribution had a standard deviation of 0.1. In Figure 3, we compare our RFGAN with GAN and unrolled GAN. From this experiment, we
confirmed the observation from (Metz et al., 2016) in that GAN easily suffered from mode collapse while unrolled GAN is effective in solving this problem.

We observed an interesting behavior in the proposed model. Those of existing studies (Arjovsky et al., 2017) and (Donahue et al., 2016) solved the mode collapse similarly to unrolled GAN; they covered the entire region of the distribution and then gradually localized the modes. However, RFGAN learned each mode first and then escaped from mode collapse when representative and features are balanced. This phenomenon is related to the characteristics of the proposed model in that the formulation of RFGAN minimizes the reverse KL divergence but, at the same time, is influenced by the representative features derived from the forward KL divergence. When the representative features no long distinguished between real and fake input, we can interpret this as the generator at this stage had the representation power of the representative features. In other words, the generator learned the entire mode as much as the representative features, and then escaped mode collapse. This is why RFGAN first responded similarly to GAN but gradually produced the entire mode.

4.2. Quantitative evaluation

Since RFGAN is built upon the baseline architecture with their suggested hyper-parameters, the input dimensionality is set at (64, 64, 3), which is acceptable for handling the CelebA and LSUN datasets. Yet, for CIFAR-10 dataset, we modified the network dimensions to fit the input into (32, 32, 3) for a fair and coherent comparison with previous studies. In addition, we drew 500 k images randomly from the LSUN-bedroom dataset for efficient training and comparison purposes.

We used two different metrics to measure 1) the visual quality and 2) the diversity of data generation. An inception score (Salimans et al., 2016) is introduced to measure the visual quality of the GANs using CIFAR-10 datasets with a larger score representative of higher quality. To evaluate the diversity of the GANs, the MS-SSIM (Odena et al., 2016) is popularly used, with which the smaller the MS-SSIM value, the better the performance in producing diverse samples.

The inception score correlates well with quality evaluation of human annotators (Salimans et al., 2016), and so is widely used for assessing the visual quality of the generated samples from GANs. Analogous to (Salimans et al., 2016), we computed the inception score for 50 k samples generated from GANs. To compare with existing GANs, we just used the DCGAN-based architecture. Yet, in order to show that the proposed algorithm is extendable to different GAN architectures, we applied our framework to other state-of-the-art GANs as well. For that, we modified their discriminators by adding representative features and trained with the same hyper-parameters initially proposed in their respective studies. From among the most recent works, we chose LSGAN (Mao et al., 2017), DRAGAN (Kodali et al., 2017), and WGAN-GP (Gulrajani et al., 2017) for comparison purposes. In the case of WGAN-GP, the generator is updated once after the discriminator is updated five times. Following the reference code, other networks are trained by updating the generator twice and the discriminator once.

Figure 4 shows plots of the inception scores as a function of epoch. We compared DCGAN, DCGAN-RF, LSGAN, and LSGAN-RF, where any extension with our representative feature is named with the postfix of RF. From this comparison, we observed that DCGAN-RF and LSGAN-RF outperformed the DCGAN and LSGAN, respectively, in terms of inception scores. Moreover, we consistently

\[1 \text{https://github.com/carpedm20/DCGAN-tensorflow}\]
found that the inception score of DCGAN-RF and LSGAN-RF grew faster than DCGAN and LSGAN, respectively, which justified that the training efficiency is improved by applying our representative feature; the proposed algorithm approached faster than the baseline GAN to achieve the same visual quality.

The other baseline GANs are DRAGAN and WGAN-GP, which have recently been proposed using the gradient penalty as a regularization term for training the discriminator. We also extended them using our representative feature and compared our extensions with the original baseline GANs (Figure 5 shows the results). Our modification still improved the inception scores although the improvement is not as significant as with DCGAN-RF and LSGAN-RF if the coefficient of gradient penalty is 10. Interestingly, we found that this coefficient played an important role in increasing the inception score: the greater the coefficient, the stronger the gradient penalty. When the gradient penalty term became stronger, training the discriminator is disturbed because the gradient update is directly penalized. When setting the coefficient of gradient penalty to 0.1 for both DRAGAN and DRAGAN-RF, we observed that the score gap between them increased, which is expected because the performance of DRAGAN approached that of DCGAN as the effect of the gradient penalty significantly decreased. However, since WGAN-GP replaces the weight clipping in WGAN with a gradient penalty, it does not satisfy the k-Lipschitz constraint with a low gradient penalty, which led to the degraded performance of WGAN-GP. Thus, it is difficult to confirm the tendency of our representative feature upon various coefficients for WGAN-GP.

By comparing the inception scores, we observed that DCGAN-RF produced the best score among the others, including existing GANs with or without our representative feature. This is analogous to the observations by (Lucic et al., 2017) in that DCGAN is the most effective model for high quality image generation. On average, our achievement over the baseline GAN is approximately 0.128, which is a considerable gain because this value is either greater than or similar to the differences between the different GANs. Moreover, the improvement is noticeable between LSGAN and LSGAN-RF.

MS-SSIM computes the similarity between image pairs randomly drawn from generated images. (Odena et al., 2016) first introduced it as a suitable measure for diversity of image generation. As discussed in (Fedus et al., 2017), it is important to note that MS-SSIM is meaningless if the dataset is already highly diverse. Therefore, we used the CelebA dataset only to compare MS-SSIM values since CIFAR-10 is composed of different classes, thus already includes highly diverse samples, and LSUN-bedroom also exhibits various views and structures, so has a diverse data distribution. For the comparison of MS-SSIM, we again chose four existing GANs as baseline algorithms (DCGAN, LSGAN, DRAGAN, and WGAN-GP), and compared them with their RFGAN variants (DCGAN-RF, LSGAN-RF, DRAGAN-RF, and WGAN-GP-RF). RFGAN considerably improved the diversity compared with the baseline GANs, and this tendency is consistent over all cases, even in the presence of the gradient penalty term; their MS-SSIM scores are summarized in Table 1.

From Table 1, the scores from LSGAN-RF, WGAN-GP-RF, and DRAGAN-RF are close to the score of the real dataset diversity, meaning that the generator is able to produce diverse samples reasonably well. Among all of them, DRAGAN-RF attained the best MS-SSIM performance by
generating the most diverse samples. A comparison of DCGAN and DCGAN-RF demonstrated the most notable improvement with the representative feature because DCGAN most frequently suffers from mode collapse, and so based on this experimental study, we can confirm that RFGAN is effective in solving this problem.

4.3. Qualitative evaluation

We compared the generated images from DCGAN and RFGAN from the same training iteration and found that our RFGAN clearly produced better results, as shown in Figure 6. Our achievement can be interpreted as speeding up the training process because the visual quality of RFGAN is similar to the results from the later iterations of DCGAN. This result is consistent with the quantitative evaluations in Figure 4.

Because we reused the training data for extracting the representative features, it is possible to question if the gain came from overfitting of the training data. To justify that our achievement is not the result of data overfitting, we generated samples by walking in latent space, which is demonstrated in Figure 7. According to (Radford et al., 2015), (Bengio et al., 2013), and (Dinh et al., 2016), the interpolated images between two images in latent space do not have meaningful connectivity (i.e. there is a lack of smooth transitions). From Figure 7, we can confirm that RFGAN learned the meaningful landscape in latent space because of it producing natural interpolations of various examples. Consequently, we conclude that RFGAN did not overfit the training data.

5. Conclusions

In this study, we develop an improved technique for stabilizing GAN training for breaking the trade-off between the visual quality and image diversity. While existing GANs explicitly add regularization terms (e.g., the gradient penalty) for improving the training stability, our approach implicitly hinder the fast growth of the discriminator updates, thus achieving stable training. To this end, we employ the representative features from representative features using an AE pre-trained with the real data. Because representative features slow down the updates of discriminator, they are effective to stabilize GAN training. As a result, we successfully improve the visual quality of generated sample and solve mode collapse. Moreover, we demonstrated that RFGAN is easily extendable to various different architectures and is robust to parameter selection.

From extensive experimental studies, we are able to justify that representative features are useful for training the discriminator of GANs. Although our representative features are trained from the AE, we believe that there could be other types of representative features or training schemes for further improving GAN performance. We hope that our work can serve as a basis for further work by employing various features or prior information to better design the discriminator.
References

Alain, Guillaume and Bengio, Yoshua. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Arjovsky, Martin and Bottou, Léon. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

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

Bao, Jianmin, Chen, Dong, Wen, Fang, Li, Houqiang, and Hua, Gang. Cvae-gan: Fine-grained image generation through asymmetric training. *arXiv preprint arXiv:1703.10155*, 2017.

Bengio, Yoshua, Mesnil, Grégory, Dauphin, Yann, and Rifai, Salah. Better mixing via deep representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 552–560, 2013.

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

Cao, Yun, Zhou, Zhiming, Zhang, Weinan, and Yu, Yong. Unsupervised diverse colorization via generative adversarial networks. *arXiv preprint arXiv:1702.06674*, 2017.

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

Chen, Lin, Rottensteiner, Franz, and Heipke, Christian. Feature descriptor by convolution and pooling autoencoders. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(3):31, 2015.

Chen, Xi, Chen, Xi, Duan, Yan, Houchoof, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2172–2180. Curran Associates, Inc., 2016.

Dinh, Laurent, Sohl-Dickstein, Jascha, and Bengio, Samy. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Donahue, Jeff, Krähenbühl, Philipp, and Darrell, Trevor. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016. URL http://arxiv.org/abs/1605.09782.

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

Fedus, William, Rosca, Mihaela, Lakshminarayanan, Bala, Dai, Andrew M, Mohamed, Shakir, and Goodfellow, Ian. Many paths to equilibrium: Gans do not need to decrease divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

Goodfellow, Ian. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

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

Kodali, Naveen, Abernethy, Jacob, Hays, James, and Kira, Zsolt. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

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.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaou. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.

Lucic, Mario, Kurach, Karol, Michalski, Marcin, Gelly, Sylvain, and Bousquet, Olivier. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

Mao, Xudong, Li, Qing, Xie, Haoran, Lau, Raymond YK, Wang, Zhen, and Smolley, Stephen Paul. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821. IEEE, 2017.
Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163, 2016.

Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), Advances in Neural Information Processing Systems, pp. 271–279. Curran Associates, Inc., 2016.

Odena, Augustus, Olah, Christopher, and Shlens, Jonathon. Conditional image synthesis with auxiliary classifier gans. arXiv preprint arXiv:1610.09585, 2016.

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

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, Chen, Xi, and Chen, Xi. Improved techniques for training gans. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 29, pp. 2234–2242. Curran Associates, Inc., 2016.

Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, ICML ’08, pp. 1096–1103, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390294.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec):3371–3408, 2010.

Warde-Farley, David and Bengio, Yoshua. Improving generative adversarial networks with denoising feature matching. 2016.

Wei, Hao, Seuret, Mathias, Chen, Kai, Fischer, Andreas, Liwicki, Marcus, and Ingold, Rolf. Selecting autoencoder features for layout analysis of historical documents. In Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing, pp. 55–62, New York, NY, USA, 2015. ACM.

Yeh, Raymond A, Chen, Chen, Lim, Teck Yan, Schwing, Alexander G, Hasegawa-Johnson, Mark, and Do, Minh N. Semantic image inpainting with deep generative models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5485–5493, 2017.

Yu, Fisher, Seff, Ari, Zhang, Yinda, Song, Shuran, Funkhouser, Thomas, and Xiao, Jianxiong. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365, 2015.

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

Zhou, Guanyu, Sohn, Kihyuk, and Lee, Honglak. Online incremental feature learning with denoising autoencoders. In Lawrence, Neil D. and Girolami, Mark (eds.), Artificial Intelligence and Statistics, pp. 1453–1461, La Palma, Canary Islands, 2012. PMLR.