A Theoretical and Empirical Examination on Recent Generative Adversarial Models

Taehee Jung, Donghyeon Ko
Department of Statistics
University of California, Berkeley
{taehee_jung,donghyeon_ko}@berkeley.edu

Abstract

Adversarial framework boosts up generation task, especially for images and videos. Our goal of this project is to explore recent adversarial generative models and their applications, and compare them in theoretically and empirically. We study vanilla Generative Adversarial Networks (GAN), deep convolutional GAN (DCGAN), Wassertein GAN (WGAN), and Cycle GAN. We tested different algorithms in extrinsic methods. What we have tested are *generation*, *interpolation*, *projection*, *arithmetic operations*, *translation*, and *completion* tasks. This is a preliminary study for our future research work. Based on our observation from this project, we propose several novel ideas for improving existing generative models or providing some interesting applications.

1 Introduction

Generative models have been actively explored especially in computer vision and natural language generation. A recent success of adversarial training [4] rapidly increase generation quality in image generation. Our goal of this project is to study the recent adversarial generative models and their applications, and compare them in theoretically and empirically.

We first study original Generative Adversarial Networks (GAN) [4] by comparing with other generative models such as variational autoencoder(VAE) [9], etc. For image generation, convolutional neural network (CNN) [10] has been easily adapted to adversarial training called Deep Convolutional GAN (DCGAN) [16].

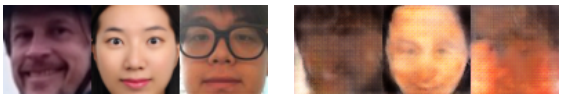
However, current GAN has some issues: To address the *mode collapsing problem* of GAN, Wassertein GAN (WGAN) [5] has been introduced for more stable and meaningful learning.

One of the interesting applications of generative models is to translate from one image to other called *image translation*. Recent works called Pix2Pix [6] and CycleGAN [18] tackle this problem by learning the pairwise translation using GAN [6] or individually learning each GAN and bridging them using encoder-decoder network [18].

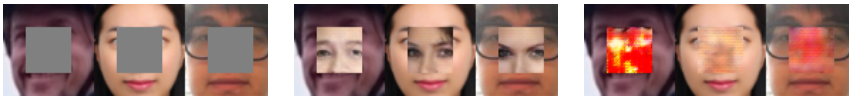
Figure 1 shows some results from our experiment (none of the outputs are borrowing from the original papers).

(a) shows projection results by training original GAN with convolutional network (DCGAN) on CelebA [13] dataset and then choosing the closest face over latent space z using contextual and perceptual loss [17]. Since the combined loss does not work well on unseen images, we crop the size of test data as small as the size of training data and test it again. However, we observed that GAN is very sensitive to size scales and unseen patterns of images. This is because GAN is learning latent space by learning implicit density instead of explicit density.

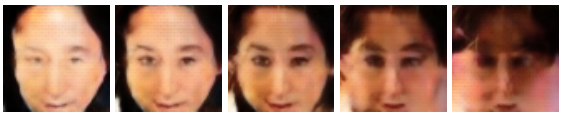
(b) shows completion task [17] by taking out center part of (unseen) input image and filling it out using generator pre-trained by DCGAN on CelebA [13]. We implement original paper's method



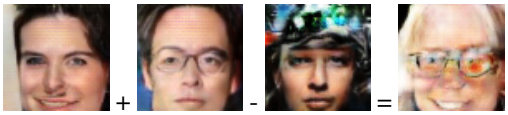
(a) **Projection** (original -> closest z in GAN)



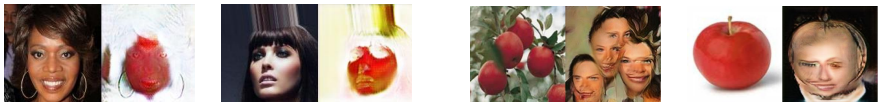
(b) **Completion** (masked -> epoch 0 and epoch 20k)



(c) **Interpolation**: Taehee -> Donghyeon



(d) **Arithmetic Operation**



(e) **Face-Apple** (left) and **Apple-Face**(right)

Figure 1: Our experiment results: (a) projection from new input faces to most closest image sampled from a latent space of GAN using Contextual and Perceptual Loss, (b) completion task, (c) interpolation from men to women class, (d) arithmetic operation, (e) face to apple or apple to face translation task. All experiment results are outcome of our experiments.

by linearly interpolating the contextual and perceptual loss but outcome wasn't good enough. Our future plan for improving this is to train our GAN using much larger and general face dataset such as OpenFace [1] instead of only celebrity faces.

(c) shows interpolation between two random z's from different classes in latent spaces of GANs. We also train DCGAN with class labels such as men, women, blond, etc as described in original DCGAN paper. The interpolation works so well which means GAN successfully models distribution of each class over latent space. Also, we tried arithmetic operation of random images by subtracting or adding z's (See (d)). It shows that arithmetic operation such as subtraction or addition leads to the expected semantic change of an image.

(e) shows image-to-image translation task trained by CycleGAN [18] with Wassertein GAN (WGAN) [5] for a better generation. We train a bit difficult task such as translating from faces to apples or vice versa. We expect a figure having a human face on apple's surface but the generated apples are more look like red faces without hair colors. We remain improvement of this problem for our future work.

Finally, we conclude and propose some ideas for improving existing generative models based on our practical experience and observation we found from this work.

2 Generative Adversarial Networks (GANs)

We study theoretical difference of recent generative models and their extensions such as GAN, WGAN, and DCGAN. Moreover, we explore interesting applications of GANs such as image-to-image translation.

2.1 Preliminary Study on Generative Models

The term 'generative model' implies any model taking training sets from probability distribution p_{data} and then learning them to get the estimate of their distribution p_{model}. Figure 2 shows a basic

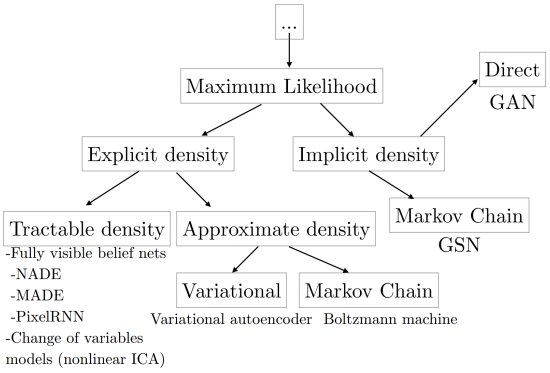


Figure 2: Taxonomy of generative models [3].

taxonomy of deep generative models learning via maximum likelihood estimations. In maximum likelihood method, p_{model} is defined with parameters θ, therefore the key is to find θ which maximizes likelihood of p_{model} with training samples $x^{(i)}$.

It can be also switched as a problem of minimizing KL divergence between data generating distribution and the model. In this case, we use an empirical distribution from sample training sets such as $\hat{p}_{data}$ since it is impossible to find the true p_{data} in practice. Equation 1 shows how to deal with the optimization problems for maximum likelihood principle.

$$\begin{aligned}\theta^* &= \arg\max_\theta \prod_{i=1}^{m} p_{model}(x^{(i)};\theta) = \arg\max_\theta \sum_{i=1}^{m} \log p_{model}(x^{(i)};\theta) \\ &= \arg\min_\theta D_{KL}\big(p_{data}(x)||p_{model}(x^{(i)};\theta)\big) \end{aligned} \tag{1}$$

The methods on the left side of the branch take the explicit density. Some of the densities may be computationally tractable. This explicit and tractable density is somewhat effective since optimization problem can be directly calculated on the likelihood of train data. However, there is a limitation since only a few of tractable densities exist. On the other hand, we can also construct explicit, but intractable density model to estimate p_{model}. In this case, the maximum likelihood should be approximated. Using stochastic approximation such as Markov chain can be one way. However, convergence of x to $p_{model}(x)$ is very slow. Furthermore it doesn't work for high-dimensional data.

Another way is the deterministic approximation, mostly dealing with variational methods. This method defines computationally tractable lower bound $\mathcal{L}$ regardless of the likelihood function used. Variational autoencoder(VAE) [9] is the most popular approach among variational methods. In VAE, we assume that unknown, latent variable z and x are generated by $p_\theta(x|z)$. This method also contains variational approximation $q_\phi(z|x)$ for posterior $p(z|x)$, which can be interpreted as a decoder. Although VAE is one of the most popular deep generative models along with GANs, the result for VAEs can be biased with weak prior and posterior distribution.

The right side branch represents methods with implicit density. Generative stochastic network (GSN) uses Markov chain which runs several times to obtain samples from p_{data}. There are obvious drawbacks on Markov chain as discussed in the previous paragraph. Generative Adversarial Network (GAN) [4] is also derived from implicit density, and directly sampled in a single step. GAN is designed to avoid several disadvantages in existing models on the taxonomy.

2.2 Generative Adversarial Models

GAN refers to the generative model with adversarial training. Unlike the other generative methods, it contains two models: generator (G) with a parameter $\theta^{(G)}$ and discriminator (D) with a parameter $\theta^{(D)}$. G is a differentiable function of random noise variable z from prior $p_z(z)$ and tries to transform z into samples which resembles a true x from training set. On the other hand, D is also differentiable function of data x but works to distinguish between the true x and the fake x from z.

While training, D tries to maximize the probability of detecting $G(z)$ as a fake, whereas $G(z)$ itself tries to make a fool of D. Therefore, this contradiction combined as one minimax problem. Equation 2 represents an objective function for GAN framework, which is coming from zero-sum game. This model will be trained by updating both $G(z)$ and $D(x)$ simultaneously. In practice, simultaneous or alternative stochastic gradient descent are used.

$$G^* = \arg\min_G \max_D V(G, D) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (2)$$

The major difference of GAN with VAE is: while VAEs can take an image into encoder and get the related vector, GANs have of course no direct way of extracting which vector of numbers of a real image would correspond to. To search for the z vector which is closest to the real image, we define a loss function that measures difference between the original image and the generated image for a given z. Then, we can find the z vector of a given image that minimizes the loss, and we call this as *projection* task.

[17] proposes a straightforward method called contextual loss which sums over absolute difference between pixel values of the original and the generated image. Using the contextual loss, they can complete missing part of images by generating with contextual loss. However, in our experiment, missing part was not so realistic looking so we used a heuristic way of by the contextual loss L_{cont} plus lambda λ times perceptual loss L_{perc}, where perceptual loss is the closest z that give most realistic image with the original image.

$$L = L_{cont} + \lambda * L_{perc} \quad (3)$$

We also test this loss for completion task in our experiment. However, the projection does not work well so we may need further clipping techniques for better reconstruction such as [12].

DCGAN [16] proposes convolutional neural network (CNN) in adversarial training. They give evidence that adversarial networks learn good representations of images for supervised learning and generative modeling. This is very useful because it allows linear arithmetic on images, directly manipulating facial features (e.g., making a person crying or wearing glasses), by simply manipulating the latent representation vector. We also test DCGAN's arithmetic operation using the projection method that we introduce earlier.

To address the *mode collapsing problem* of GAN, Wassertein GAN (WGAN) [5] has been introduced for more stable and meaningful learning. For our image translation task, we use WGAN instead of vanilla GAN.

2.3 Image-to-Image Translation

One usage of GAN is to translate one image to other image called image translation task. Two recent approaches are popular to solve the image-to-image translation task using GANs. While Pix2Pix [6] requires a pairwise image dataset for training, CycleGAN [18] only requires two different classes of images by individually training each class of images and learning their transition using encoder-decoder model. In this project, we test CycleGAN for the image-to-image transition task.

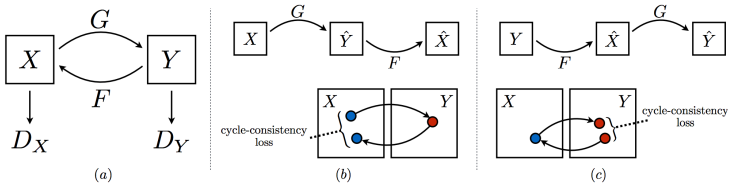


Figure 3: Framework for CycleGAN: (a) It contains two mapping functions G,F and each of them has a separate discriminator D_X and D_Y. While D's are working to separate real images from translated images, G and F try to make fake images as realistic as possible. (b) forward cycle-consistency loss to make G(F(y)) $\approx$ x and (c) forward and backward cycle-consistency loss to make F(G(x)) $\approx$ y . [18]

Main idea of these models is to learn a mapping function G(x) which maps image x to the output image y based on the GAN approach. While Pix2Pix only considers one-sided mapping so that it needs pairwise training examples $(x_i, y_i)_{i=1}^N$, CycleGAN contains two mapping functions: G: x $\rightarrow$ y

and F: y $\rightarrow$ x. Transition on both hand is successful only in case that G and F are an inverse to each other. To reflect this concept, cycle-consistency loss is included in an objective function 4.

$$G^*, F^* = \arg\min_{F,G} \max_{D_x, D_y} \mathcal{L}_{\text{GAN}}(G, D_Y, Y, X) + \mathcal{L}_{\text{GAN}}(F, D_X, X, Y) + \lambda \mathcal{L}_{cyc}(G, F) \quad (4)$$

$$\text{where } \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x))]$$

$$\text{and } L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] + E_{y \sim p_{data}(y)}[||F(G(y)) - y||_1]$$

Using cycleGAN, we perform image-to-image transitions with 2 unpaired image groups from similar types of category and also from very different categories.

3 Experiments

In the experiment, we use several dataset for different tasks: For DCGAN training, we use celebrities' face dataset called CelebA [13] and hand written digits dataset called MNIST [11] for testing GAN and other tasks. We also use photos of Professor Shewchuk[1] and ourselves for testing. For CycleGAN training, we use the dataset that the original paper has used[2].

We refer to the codes from github to initialize DCGAN[3] and CycleGAN[4]: For DCGAN, we use the reference code to generate models and implement projection, completion, interpolation and arithmetic operation task. For CycleGAN, we mainly use the reference code but partially amend it to load our experiment data.

To validate the different GANs in multiple perspectives, we test the algorithms in extrinsic methods as follows: general *generation* quality check, *Interpolation*, *Projection and Completion*, *Operation*, and *image-to-image translation* between different classes of images such as faces and apples.

3.1 Adversarial Training and Generation Quality

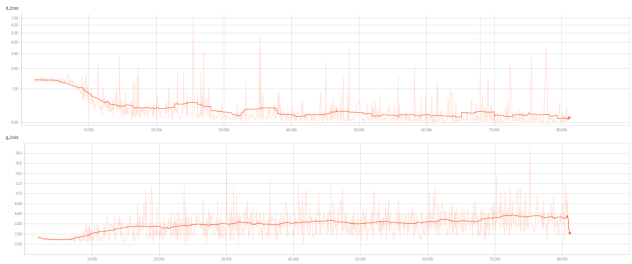


Figure 4: Discriminator (top) and generator (bottom) loss during training on CelebA

Figure 4 shows loss of discriminator (top) and generator (bottom) during training on CelebA dataset. The actual losses are bit smoothed by averaging to effectively show the trend. As described in earlier section, we could observe that GAN minimizes discriminative loss while maximizes generator loss. We used CNN with GAN (aka DCGAN) with 108 width and height for facial images, cropping, 25 epochs of training with 0.0002 learning rate for Adam [8] optimizer, and 32 batch size with 3 channels for image colors.

For each epoch of training, we generate random images for both CelebA and MNIST dataset (See Figure 5 top for CelebA and bottom for MNIST). We could observe that images are getting accurate and clear as the number of epochs increases.

3.2 Interpolation and arithmetic operation

Figure 6 shows our interpolation results. Since we also use labels on CelebA dataset, we choose random z from two different classes such as between men and women or between blond and brown,

[1] https://people.eecs.berkeley.edu/~jrs/

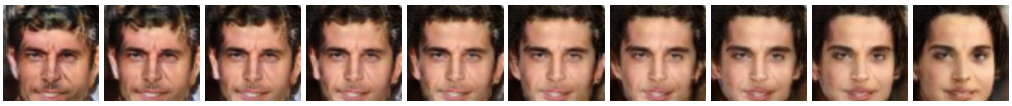
[2] https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/

[3] https://github.com/carpedm20/DCGAN-tensorflow

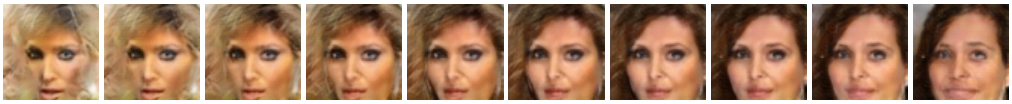
[4] https://github.com/hiwonjoon/cycle-gan-tf

(a) **CelebA (epoch 0/10/20)** (b) **MNIST (epoch 0/10/20)**

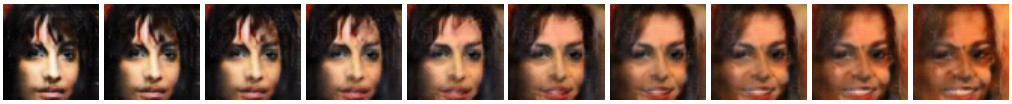
Figure 5: Random generation on CelebA (left) and MNIST (right) dataset over epochs at 0/10/20.



(a) **Men -> Women**



(b) **Blond -> Brown**



(c) **White -> Black**

Figure 6: Interpolation tasks on CelebA dataset

or between white and black. We randomly choose two images from each class, interpolate the internal z's in the latent space by size 10 and sample each image. Surprisingly, we could obtain how man changes to woman, how blond hair changes to brown, and how white woman changes to black woman. This interpolation results show that GAN is smoothly projecting images over continuous space (z).

The next step using this continuity is doing arithmetic operation (z1+z2-z3) of latent variables such as word embedding [14] does. Figure 7 shows three examples and the results looks to make sense. First example starts with z1 from woman, add z2 for man with glasses, and subtract z3 for young man with colorful hat. Final answer looks like an old man with glasses and gray hair. In the second example, we start with z1 for woman, add z2 from old man, then subtract z3 from another woman. In final we get another old man. The last example starts with woman's side face with brown hair, close her mouth, add a blond woman smiling and looking in front then subtract woman looking in front, closing her mouth and having brown hair. Final image from operation looks like a woman's side face with smiling and light hair.

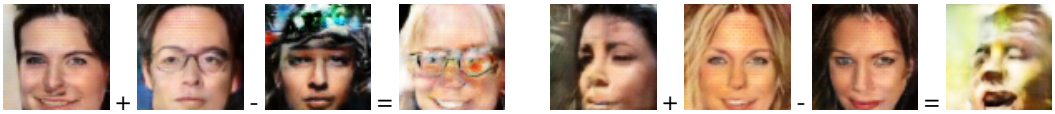
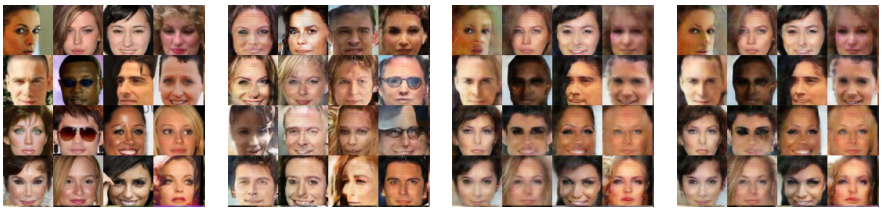


Figure 7: Arithmetic operation between random zs. First z + second z - third z.

3.3 Projection and Completion

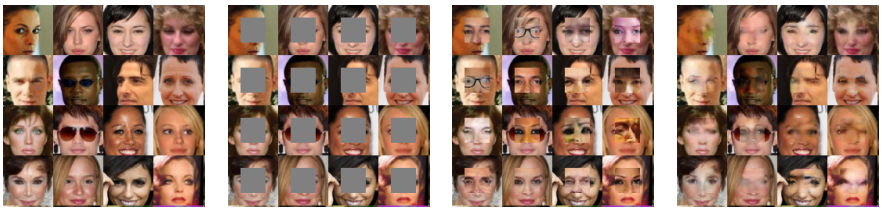
Figure 8 shows projection results from (a) original random testing images after epochs at (b) 0, (c) 1000, and (d) 30000 by training DCGAN on CelebA [13]. Then, we choose the closest face over latent space z using contextual and perceptual loss [17]. We used 0.1 weight for perceptual loss, and 0.8 momentum with 0.002 learning rate. After 30000 epochs of training for minimizing the combined loss, we could obtain almost similar images like original testing images.

Figure 9 shows completion results from (a) original random testing images (b) masked by gray square at center, and epochs at (c) 0 and (d) 20,000. As same as projection task, we train original DCGAN on CelebA [13] and then choose the closest face over latent space z using contextual and perceptual



(a) **Original** (b) **Epoch 0** (c) **Epoch 1,000** (d) **Epoch 30,000**

Figure 8: Projection results with contextual and perceptual loss.



(a) **Original** (b) **Masked** (c) **Epoch 0** (d) **Epoch 20,000**

Figure 9: Completion results with contextual and perceptual loss.

loss [17]. We used same hyperparameter as projection task. After 20,000 epochs of training, we could recover the masked part of images as same as the original images.

Table 1: Experiment with contextual (only), perceptual (only) and interpolation with both of them.

Contextual (only)	Perceptual (only)	Contextual+Perceptual (lam=0.1)
274.51	351.21	254.19

For both projection and completion task, we could obtain comparable loss values as the original paper [17]. Table 1 shows our final loss for completion tasks on our unseen test images. We obtain the interpolation weight (lam) by running multiple runs from 0.0 to 1.0. The combined loss with 0.1 weight achieves the best value compared to contextual only and perceptual only loss. We also test these two tasks on unseen images such as our faces and professor's face 1 but the results were not good as the testing images. We observe that GAN is very sensitive to image scaling size and new patterns from unseens images. For example, CelebA dataset has very small number of Asian faces so it couldn't recover them well.

3.4 Image-to-Image Translation

Figure 10 shows our image-to-image translation results trained on three different pairs of images: (a) Vangogh images to general scenery photos or vice versa, (b) aerial map to Google style map or vice versa, and (c) face to apple or vice versa. If two pairs of images are in similar types of classes such as scenery photos or map like photos, the translation works amazingly well. However, as shown in the introduction, transition between two very different categories such as faces and apples is not working well. Apples from faces are not accurate because originally they have different features so some features such as hairs are just ignored. But, common features such as round shape of faces and apples are pretty well aligned.

4 Conclusion

So far, we have studied recent GANs and tested state-of-art techniques derived by GANs. We show many good results but there are also some bad examples because of the limitations of the methods. We will supplement them and continue on the research mentioned in the next section.

(a) **Vangogh <-> Photo**

(b) **Aerial <-> Map**

(c) **Face <-> Apple**

Figure 10: Image transition using CycleGAN.

5 Future Works

5.1 Multi-modal Style Transfer

During this project, we also study some recent works on style transfer on images. Our proposed method is to transfer styles not only between images but also between texts. In order to align text with images, we consider using image captioning tasks such as [7]. For example, if two images are given, we can generate captions (explanation) for the images. Then, when we transfer painting styles from one image to other image, we also align the transition with textual writing styles from one caption to other caption.

5.2 Expecting Children Future Face Generation using GAN and CycleGAN

We are currently working on using CylcleGAN for detecting the kinship patterns based on facial features [15]. We interpolate the father and mother photos using our projection technique and then train CycleGAN between the interpolated parents (father+mother) image and children's image. Once it is trained, any given father and mother photos we could generate their future children's faces (hopefully). This work will be uploaded in Archive soon.

5.3 Evaluating Generated Images

While working on the project, we wondered why is GAN so popular these days. There can be many plausible reasons, but one possible reason that we were particularly interested in was the quality of generated images. We wanted to know whether the quality of the images generated by GAN is better than the quality of the images generated by other methods and whether there is a variation of GAN that generates better images than other variations of GAN. To perform this task, we need quantitative measure to evaluate the results of the models.

Due to the lack of time to implement several models, we couldn't actually perform this task. However, we will briefly explain how we would have done this task if time permitted [2]. We let images $G = \{g_i\}_{i=1}^m$, $g_i \in R^d$, real images $R = \{r_i\}_{i=1}^n$. First, we compute the singular value decomposition U_G, Σ_G, V_G and replace each image g_i with the image's principal component coefficients g_i'. Then each image is replaced with the Gaussian centered around $g_i', N(g_i', \Sigma_G)$. We truncate g_i' and Σ_G properly. We do the same process to create a Gaussian mixture model for the set R. Then we approximate KL divergence which will be the quantitative measure by:

$$\pi(i) = \arg\min_j KL[N(r_i', \Sigma_R)||N(g_j', \Sigma_G)]$$

$$\text{Approximate KL divergence} = \frac{1}{n}\sum_{i=1}^{n} KL[N(r_i', \Sigma_R)||N(g_j', \Sigma_G)] \tag{5}$$

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. 2016. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science.

[2] Steven Basart and Dan Hendrycks. 2017. A quantitative measure of generative adversarial network distributions.

[3] Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* .

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

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

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

[7] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pages 3128–3137.

[8] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

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

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.

[11] Yann LeCun, Corinna Cortes, and Christopher JC Burges. 1998. The mnist database of handwritten digits.

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

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

[15] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. 2015. Tri-subject kinship verification: Understanding the core of a family. *IEEE Transactions on Multimedia* 17(10):1855–1867.

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

[17] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. 2016. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539* .

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593* .