Resembled Generative Adversarial Networks: Two Domains with Similar Attributes

Duhyeon Bang
duhyeonbang@yonsei.ac.kr
Hyunjung Shim
kateshim@yonsei.ac.kr
School of Integrated Technology, Yonsei University, South Korea

Abstract

We propose a novel algorithm, namely Resembled Generative Adversarial Networks (GAN), that generates two different domain data simultaneously where they resemble each other. Although recent GAN algorithms achieve the great success in learning the cross-domain relationship [9, 19, 22], their application is limited to domain transfers, which requires the input image. The first attempt to tackle the data generation of two domains was proposed by CoGAN [10]. However, their solution is inherently vulnerable for various levels of domain similarities. Unlike CoGAN, our Resembled GAN implicitly induces two generators to match feature covariance from both domains, thus leading to share semantic attributes. Hence, we effectively handle a wide range of structural and semantic similarities between various two domains. Based on experimental analysis on various datasets, we verify that the proposed algorithm is effective for generating two domains with similar attributes.

1 Introduction

Generative adversarial networks (GANs) are capable of producing sharp and realistic images by learning the generative process, instead of explicitly estimating the data distribution with variational bounds or strict model constraints. GANs [5] is composed of two networks, discriminator and generator, and they adversarially compete each other to approximate P_{data} using P_{model}: the discriminator distinguishes real samples from fake samples produced by the generator, while the generator aims to create the sample as real as possible so that the discriminator cannot recognize it as the fake sample. The objective function of this adversarial learning process in [5] is defined by the following minimax game,

$$\min_G \max_D \mathbb{E}_{\mathrm{x}\sim P_{\mathrm{data}}}[\log(\mathrm{D}(\mathrm{x}))] + \mathbb{E}_{z\sim P_z}[\log(1-\mathrm{D}(\mathrm{G}(\mathrm{z}))],$$

where $\mathbb{E}$ denotes expectation, x and z are random variables for data and latent vector, where their probability distributions are P_{data} and P_z, respectively.

Most GAN algorithms learn an unidirectional mapping function from P_z to P_{data} for a single domain. Unlike those, our algorithm learns two mapping functions for two domains simultaneously; one associates P_z to P^{x}_{data}, and the other maps the same P_z to P^{y}_{data}. Throughout this paper, we denote two different domains by X and Y, respectively. When x and y are samples generated from the same latent z, we aim to accomplish two objectives; 1) two data

© 2018. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

obey their own data distribution, and 2) two data hold shareable characteristics as similar as possible. For example, faces of human and those of cat represent different species, technically different domains. Hence, they have different shapes and structures. However, their posture, hair color, or facial expression can be similar in both domains, thus those attributes can be regarded as shareable characteristics. Then, our research goal is to generate a pair of human face and cat face that faithfully produce their domain characteristics, and at the same time both faces show the similar attributes such as pose, hair style, or facial color. Note that our algorithm is categorized as an unsupervised GAN because we do not rely on correspondences between two different domains. Also, our approach does not condition on additional input data (i.e., unconditional GAN). Thus, our algorithm is categorized into an unsupervised unconditional GAN for generating two domain data simultaneously.

Recent studies learn the relationship between two different data domain for domain transfer. They include cycleGAN [22], DiscoGAN [9], and dualGAN [19]. Because they aim to establish an image-to-image translation, they require to have the input image as the given condition so to generate the output image in different domain. This problem is inherently analogous to the problem of conditional GAN; it is a different problem from unconditinoal GAN where the data is generated from P_z. CoGAN [10] made the first attempt toward the unsupervised unconditional GAN for generating two domains, as same as our study. They formulate this problem by learning the joint distribution $P^{x,y}_{data}$. Suppose the joint probability distribution is factorized as $P^{x,y}_{data} = P^x_1 \cdot P^y_2 \cdot P^{x,y}_3$. CoGAN assumes that $P^{x,y}_3$ is related to high-level semantics while P^x_1 and P^y_2 are related to low-level details. Based on this assumption, they suggests a new GAN architecture of two generators. To model $P^{x,y}_3$, first several layers of two generators are coupled by a weight-sharing constraint. The last remaining layers for both generators are designed to learn P^x_1 and P^y_2, respectively. It is because the first layers decode high-level semantics and the last layers decode low-level details in the generator. Although each generator is trained with samples for a single data domain, two generators are enforced to share high-level representations during training because of the shared layers. This weight-sharing constraint works well when there is the high structural similarity between two domains. However, with the low structural similarity, the network constraint for CoGAN is too restrictive to achieve the factorization; two generated samples may not show similar attributes.

Unlike CoGAN, the proposed GAN, namely Resembled GAN, does not explicitly design the network architecture to enforce structural similarity. Instead, our approach employs the feature statistics as an additional constraint, thus it is naturally more flexible to handle a wide range of structural similarities between two data domains.

The main contribution of this study is to define a new objective function of discriminators that leads generators to model the joint distribution, $P^{x,y}_{data}$. To factorize this joint distribution, we propose a feature statistic matching algorithm. Suppose that we derive a feature space where all samples are representative. On this feature space, we assume that the feature distribution of all training data for each domain forms a multivariate Gaussian distribution. After that, we regard a mean vector of each Gaussian as the independent component (i.e. domain specific characteristics), associated with P^x_1 or P^y_2. On the other hand, the covariance matrix represents the dependent component (i.e. shareable attributes), associated with $P^{x,y}_3$. Under this assumption, we enforce that two feature distributions have similar feature covariance matrices, effectively leveraging the covariance of two feature distributions.

Using our algorithm, different levels of structural similarity is accounted by the feature covariance; lower the similarity, greater the difference of covariance matrices. As the results,

even two domain data are structurally quite different, we can maintain the quality of data generation as well as the attribute similarity.

2 Background

We categorize recent studies handling multiple data domains using GANs based on 1) supervised versus unsupervised approach, and 2) conditional versus unconditional approach.

Supervised Conditional GAN (*e.g.*, Image-to-Image translation): Pix2Pix [8] and BicycleGAN [23] propose Image-to-Image translation techniques, which transforms the image of the input domain to the image of the target domain. To learn such transformation, they utilize a set of paired images as training data, and they employ the mapping constraint; the transformed image should match the paired image. Finally, they combine a GAN loss with a traditional loss (*e.g.*, L_1 or L_2) and learn a transformation function that maps X to Y domain. Moreover, they utilize the input image as priors for training the generator and discriminator. The main difference between two studies is a complexity of mapping functions; pix2pix aims to learn a one-to-one mapping while BicycleGAN learns a one-to-many mapping.

Unsupervised Conditional GAN (*e.g.*, Learning domain transfer): CycleGAN [22], DiscoGAN [9], and DualGAN [19] are domain transfer algorithms using unpaired two domain images. Because there are no correspondences between images from two domains, they develop two mapping functions (i.e., forward and inverse mapping) and utilize the cycle consistency loss; the sequential operation of forward and inverse mapping should result in the identity mapping. They produce plausible results in learning cross domain relationship. However, they are incapable of generating other domain data without the input.

Unsupervised Unconditional GAN (*e.g.*, Learning joint distribution): CoGAN [10] first suggest the unsupervised unconditional GAN for generating two domain data from P_z at the same time, using unpaired two domain data. This is achieved by enforcing a weight-sharing constraint that restricts the generator capacity and favors a joint distribution solution over a product of marginal distributions. Later, the idea of weight sharing was extended to multiple domain translations by Lucic *et al*. [11]. However, this study is categorized as unsupervised conditional GAN because it requires input images for generating corresponding translated images.

3 Unsupervised unconditional Resembled GAN

Our goal is to generate paired images corresponding to two different domains with unpaired training sets. Given two different domain distributions, we aim to train two generators that can faithfully reproduce original characteristics of each domain data distribution (i.e. statistically independent component), and at the same time, retain shareable characteristics (i.e. dependent component of joint distribution) as similar as possible. For that, we define the feature space that represents both domain characteristics using an encoder of a pre-trained autoencoder (AE), and then match two covariance matrices of two feature distributions derived through the encoder.

MGGAN [1] first introduced the idea of inducing the generator to possess specific manifold characteristics using a guidance network. Their guidance network leads the generator to learn Forward KL divergence by matching the feature distribution of fake images to that of real images. In this way, they effectively solve a mode collapse problem without sacrificing

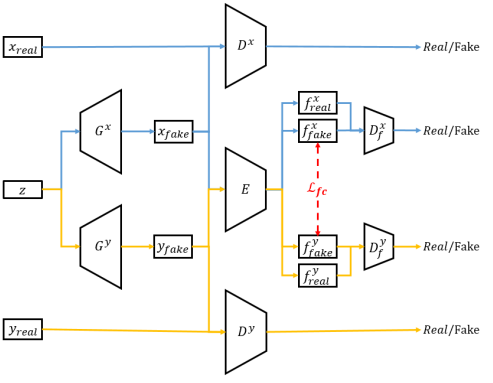

Figure 1: Architecture of Resembled GAN. X and Y represent two domain data. x_{real} (y_{real}) and x_{fake} (y_{fake}) are the sample of P_{data} and P_{model}, respectively. z is latent vector. E, G, and D are the encoder, the generator, and the discriminator network. The superscript of G^{x} (G^{y}) and D^{x} (D^{y}) indicates the domain and the subscript of $\mathrm{D}^{\mathrm{x}}_f$ ($\mathrm{D}^{\mathrm{y}}_f$) indicates the feature space.

the image quality of baseline GAN. Inspired by this success, Resembled GAN employs the idea of guidance networks into our problem. That is, we construct a new GAN architecture that two generators produce samples that represent their own domain characteristics and shareable attributes of each other.

We first define two discriminators for each generator; D^{x} (D^{y}) distinguishes the real distribution from the fake distribution while $\mathrm{D}^{\mathrm{x}}_f$ ($\mathrm{D}^{\mathrm{y}}_f$) distinguishes the real feature distribution from the fake feature distribution. The feature space is defined by an encoder, E. Then, we introduce a new loss term $\mathcal{L}_{\mathrm{fc}}$ that represents the norm between the feature covariance matrix of x and that of y. We refer this term as the feature covariance constraint, which enables to implicitly learn the joint distribution. Figure 1 visualizes the network architecture for our Resembled GAN. Its overall objective function is summarized as follow. Firstly, the objective of four discriminators (i.e., D^{x}, D^{y}, $\mathrm{D}^{\mathrm{x}}_f$, and $\mathrm{D}^{\mathrm{y}}_f$) is

$$\min_{\mathrm{D}^{\mathrm{x}},\,\mathrm{D}^{\mathrm{y}},\,\mathrm{D}^{\mathrm{x}}_f,\,\mathrm{D}^{\mathrm{y}}_f} \mathop{\mathbb{E}}_{x\sim P^{\mathrm{x}}_{data}} \left[\log \mathrm{D}^{\mathrm{x}}(x) + \log \mathrm{D}^{\mathrm{x}}_f(\mathrm{E}(x))\right] + \mathop{\mathbb{E}}_{y\sim P^{\mathrm{y}}_{data}} \left[\log \mathrm{D}^{\mathrm{y}}(y) + \log \mathrm{D}^{\mathrm{y}}_f(\mathrm{E}(y))\right]$$
$$+ \mathop{\mathbb{E}}_{z\sim P_z} \left[\log\left(1-\mathrm{D}^{\mathrm{x}}(\mathrm{G}^{x}(z))\right) + \log\left(1-\mathrm{D}^{\mathrm{y}}(\mathrm{G}^{y}(z))\right) + \log\left(1-\mathrm{D}^{\mathrm{x}}_f(\mathrm{E}(\mathrm{G}^{x}(z)))\right) + \log\left(1-\mathrm{D}^{\mathrm{y}}_f(\mathrm{E}(\mathrm{G}^{y}(z)))\right)\right]. \tag{1}$$

To update the parameters of two generators (i.e., G^{x} and G^{y}), we optimize the following objective function.

$$\min_{\mathrm{G}^{\mathrm{x}},\,\mathrm{G}^{\mathrm{y}}} -\mathop{\mathbb{E}}_{z\sim P_z} \left[\log\left(\mathrm{D}^{\mathrm{x}}(\mathrm{G}^{x}(z))\right) + \log\left(\mathrm{D}^{\mathrm{y}}(\mathrm{G}^{y}(z))\right) + \log\left(\mathrm{D}^{\mathrm{x}}_f(\mathrm{E}(\mathrm{G}^{x}(z)))\right) + \log\left(\mathrm{D}^{\mathrm{y}}_f(\mathrm{E}(\mathrm{G}^{y}(z)))\right)\right] + \omega\mathcal{L}_{fc}, \tag{2}$$

$$\mathcal{L}_{fc} = \left\| \mathrm{E}(\mathrm{G}^{x}(z)) - \mathbb{E}_{x\sim P^{\mathrm{x}}_{data}}[\mathrm{E}(x)] \,,\, \mathrm{E}(\mathrm{G}^{y}(z)) - \mathbb{E}_{y\sim P^{\mathrm{y}}_{data}}[\mathrm{E}(y)] \right\|_1 \tag{3}$$

where ω serves the weighting factor for $\mathcal{L}_{fc}$. We update the discriminators and the generators alternatively, and two discriminators for each domain, D^{x} (D^{y}) and $\mathrm{D}^{\mathrm{x}}_f$ ($\mathrm{D}^{\mathrm{y}}_f$), are trained independently.

3.1 Analyzing the feature distribution

To utilize the statistics of feature distribution as constraints, we first develop the feature space where all samples from two domains are analyzed. To compare feature statistics from two

domains, we should ensure that this feature space should be representative for all samples from both domains. Suppose that they represent only the major mode in each domain, or are biased toward one of two domains in feature space. In such a case, generators suffer from mode collapse or fail to learn the other domain characteristic. To cope all data from both domains, we utilize an encoder from a pre-trained AE. Because AE aims to reconstruct all training samples, the encoded features from AE faithfully represents the data distribution [16]. Using all samples from both domains as training set, we can ensure that the AE can encode all samples into the same feature space; two feature distributions from both domains can be comparable. Furthermore, we modify the AE to the denoising AE in order to improve the robustness of model as a feature extractor [18]. Finally, we pre-train and fix the parameters of the AE during GAN training. In this way, we maintain the representative power of feature space defined by its encoder.

3.2 Feature covariance constraint, $\mathcal{L}_{fc}$

Our key idea is to enforce the feature covariance constraint for GAN training, implicitly learning the joint distribution. The similar idea has been discussed in the previous study, Snell *et al*. [17] for few-shot learning. Assuming the feature distribution on embedding space as a multivariate Gaussian distribution, they claim that the mean of real feature distribution represents the identity attribute of the class while the covariance of that represents intra-class variation (*e.g*., shareable attributes). Inspired by this observation, we regard the mean of feature distribution as domain identity attribute, and the covariance of that as shareable attributes. From the similar observation, several studies for low-shot learning augment the data of long-tailed classes by referencing the feature covariance of rich class data. Given a single or few training image, they manipulate their features of training set to generate additional training data. For example, Yin *et al*. [20] transfer the principal components from regular classes to tail classes so to increase their intra-classes variance.

4 Evaluation

In this section, we first analyze the performance of the Resembled GAN by 1) adapting various domains, 2) conducting the image reconstruction, and 3) generating by the latent space walking. Then, we compare our model with CoGAN, which is a baseline algorithm and evaluate how well their generation retains the semantic similarity between two domains. Also, we quantitatively evaluate the generation quality both in terms of diversity and image quality.

To confirm whether each model can handle a wide range of structural similarities across domain, we experiment with two scenarios.

1. High structural and semantical similarity: We divide CelebA [12] dataset into two domains using its attribute labels; such as gender, hair colors or with/without glasses. This scenario is relatively easy because any pair of domains have high structural similarity (i.e., all are human faces) with the precise alignment.

2. Low structural and semantical similarity: We choose the CelebA dataset and Cat head dataset [21] to construct the problem of handling significantly different two domains. This scenario is relatively hard because human and cat face has low structural similarity with the poor alignment.

D, E	G	D_f
5×5 64 conv, ↓, BN, leaky ReLU	4 - 4 - 1024 fc, BN, ReLU, reshape ($4 \times 4 \times 1024$) ★	1024 fc, leaky ReLU
5×5 128 conv, ↓, BN, leaky ReLU ★	5×5 512 conv, ↑, BN, ReLU ★	1024 fc, leaky ReLU
5×5 256 conv, ↓, BN, leaky ReLU ★	5×5 256 conv, ↑, BN, ReLU ★	dimension of P_z fc
5×5 512 conv, ↓, BN, leaky ReLU ★	5×5 128 conv, ↑, BN, ReLU ★	
D : 1 fc, sigmoid ★	5×5 3 conv, ↑, Tanh	
E : dimension of P_z fc		

Table 1: Architectures for the networks that comprise CoGAN and Resembled GAN. fc and conv means a full connected layer and a convolutional layer respectively. ↑ and ↓ represent up- and down-sampling respectively. BN denotes batch normalization [7]. ReLU and Tanh denote rectified linear unit and hyperbolic tangent activation function, respectively. ★ indicates shared layers of CoGAN.

For fair comparison, we design the architecture of discriminator and that of generator for Resembled GAN and CoGAN based on DCGAN [15]; for that, we crop and resize all dataset into $64 \times 64 \times 3$. The architecture of manifold discriminators, D_f^x and D_f^y, follow that of MGGAN. Table 1 demonstrates the architecture design in detail. We crop the facial region of the cat head using facial key points provided by the original cat head dataset. Among 10k cat images, 9k images are used for training while 1k images are used for testing. For improving the training stability, we increase the training dataset to 90k using the affine transform based data augmentation. During training, we randomly draw 90k images from the CelebA dataset to match the number of Cat head data. To implement CoGAN, except the last layer of generator and the first layer of discriminator, the rest of layers from each domain network is tied for a weight-sharing constraint. We set $\omega = 1$ for all experimental evaluations.

4.1 Evaluating the semantic similarity between two domains

First, we evaluate the Resembled GAN with various paired domains: with/without glasses and black/blond hair styles. Our generation results are summarized in Fig. 2. These generated images present the domain characteristics reasonably well such as the presence of glasses or color of hair while they hold shareable attributes such as the rest of facial structure.

Resembled GAN is capable of reconstructing real data by introducing a small modification to the network. The image reconstruction has originally been demonstrated in MGGAN. That is, we add the three layers of fully connected network to map P_f to P_z. To perform the image reconstruction with CoGAN, they should conduct the additional optimization for searching a latent vector corresponding to real data. Unfortunately, this optimization is error prone and unreliable because it should solve inverse generation process, which is extremely non-linear and complicated like the generation process. Using Resembled GAN, the reconstruction can be easily formulated as a part of generative model, thus the reconstruction results are more plausible than optimization based reconstruction. Fig. 3 shows our reconstruction results. Each of the first, second, and third column are real images, reconstructed images and generated images in the other domain.

To verify whether data generation is the results of data memorization or not, we generate samples by latent space walking. The generated images from interpolated latent vectors between two specific vectors do not have meaningful connectivity if the generator just memorize the dataset, such as lack of smooth transitions or fail to generation [2, 15]. From semantically smooth interpolation results shown in Fig. 4, we conclude Resembled GAN reproduces the data distribution without memorization. More interestingly, we observe that latent walking in two domains demonstrates semantically similar. For example, in the middle

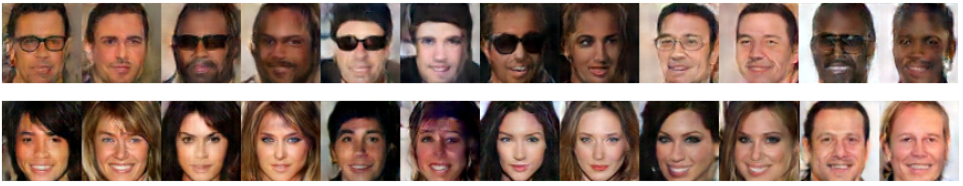

Figure 2: Generation with high structural and semantical similarity dataset; with/without glasses(top), black/blond hair(bottom). Each of odd and even columns are resembled image generated from same random noise vectors.

Figure 3: Reconstruction results. Each of the first, second, and third columns are real test image, a reconstructed image, and a resemble image of the corresponding domain.

row of Fig. 4, the smooth transitions of facial poses are quite similar in both domains. We have the same observation consistently over various examples.

Finally, we qualitatively compare Resembled GAN with CoGAN. For fair comparison, we draw samples at the same iteration of training, 35k. The generated images are shown in Fig. 5; the left half are from CoGAN, and the right half are from Resembled GAN. Each of odd and even column images are paired, each generated from G^x and G^y from the same random vectors, $z \sim P_z$.

When generating the different gender, the quality of CoGAN and Resembled GAN are nearly the same. Both algorithms retain each domain characteristic clearly (male and female) while keeping shareable attributes; such as smile, skin color or background. An interesting observation is that some attributes associated with the domain characteristics, such as mustache, are excluded from shareable features automatically by networks decision. On the contrary, when handling human and cat faces, Resembled GAN clearly is better than CoGAN, especially how well two generated images share common attributes. Because CoGAN generates images with the weight-sharing constraint, several results can also match the face orientation. However, they do not share hair color, facial shape or eye shapes, which are properly modeled shareable attributes in Resembled GAN. The generated pairs from Resembled GAN can match the facial orientation, background color, hair color, skin tone, facial shape or eye shape (e.g., oval or line-shape).

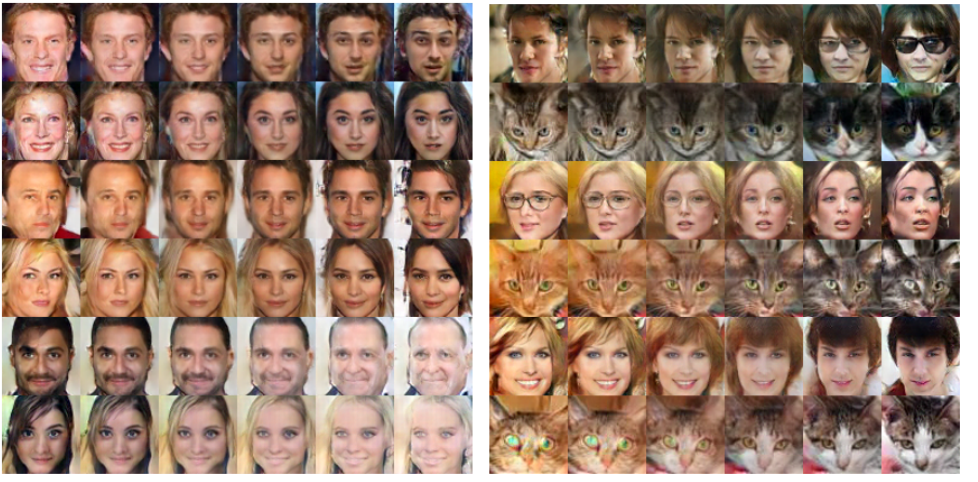

Figure 4: Interpolation results. Each of odd and even rows are pair images generated from same interpolated latent vectors.

Although CoGAN aims to match high-level semantics, weight-sharing constraint just considers the structural similarity in substance. Thus, their approach is less effective when structural features between two domains are substantially different. Unlike CoGAN, Resembled GAN constrains the feature statistics. Hence, the strength of constraint is automatically determined by the discrepancy of two domains; the bigger the difference, the stronger the constraint. For this reason, our model is more robust to handle various levels of structural and semantic similarities between two domains.

4.2 Quantitative Evaluation

Using existing evaluation metric, it is hard to quantitatively evaluate the semantic similarity of two domains. Instead, we utilize two general metrics for evaluating GANs; one is MS-SSIM for measuring image diversity (*e.g.*, the lower the MS-SSIM, the higher the diverse [3, 14]) and the other is a Fréchet distance (FID) for measuring visual quality. (*e.g.*, the higher the FID score, the higher the quality [6, 13])

When training the generators of two domains by keeping the shareable attributes, the generation process tends to increase its intra-class variations because it learns the feature covariance from both domains. As a result, the diversity of generated images in each domain becomes higher than that of real data if generators are influenced by the shareable attributes from the other domain. We observe the same phenomenon in our experiment. For example, although faces of cat in training dataset do not possess red hairs, Resembled GAN can generate cat with red hairs as shown in the last row in Fig. 5. This observation consistently holds in the quantitative evaluation summarized in Table 2. All scores are the average MS-SSIM repeated five times for each model and dataset. Resembled GAN achieves greater diversity (i.e., lower MS-SSIM) than all real dataset and CoGAN. From these results, we justify that Resemble GAN possesses the representation power for generating the wide range of attributes, more flexible to model various attributes.

Because the trade-off relationship between visual quality and sample diversity is a well-known issue in GAN training [4], we also verify whether our achievement in image diversity

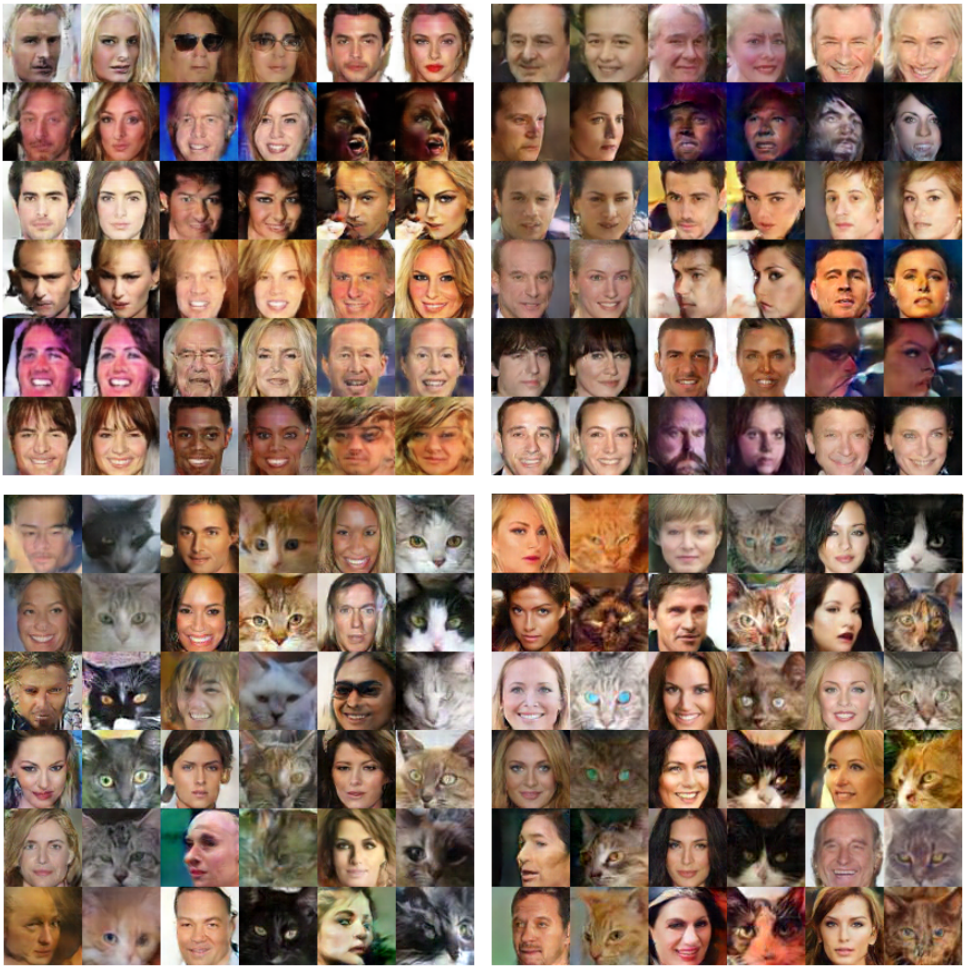

Figure 5: Comparison between CoGAN (Left) and Resembled GAN (Right) using two cases of dataset: one (top) utilize male and female of CelebA; the other (bottom) utilize celebA and cat face dataset.

Metric	Model	Male	Female	Human	Cat
MS-SSIM (mean)	Real-dataset	0.3558	0.4214	0.3897	0.2134
	Coupled GAN	0.3584	0.4351	0.3961	0.2123
	Resembled GAN	0.3392	0.4090	0.3324	0.2069
FID score (mean ± std)	Coupled GAN	34.55±2.45	29.59±3.34	38.74±3.32	31.45±4.21
	Resembled GAN	36.35±2.21	37.33±4.02	41.89±3.21	33.89±4.59

Table 2: Comparison of the sample diversity and quality using MS-SSIM (mean) and FID score (mean and standard deviation), respectively. The lower MS-SSIM, the higher diversity. The lower FID score, the higher quality.

is the result of sacrificing the image quality. The FID scores in Table 2 show that Resembled GAN reports a slightly lower FID than CoGAN in average. However, because the one standard deviation of FID from CoGAN overlaps with Resembled GAN, the statistical difference is not significantly meaningful.

5 Conclusion

This paper introduces Resembled GAN that generates a pair of images from two domains with similar attributes. The objective of our study is different from those of domain transfer techniques in that we deal with unsupervised and unconditional approach to generating two domains simultaneously. While existing method for the same objective, CoGAN, explicitly enforces the structural similarity between two domains, we induce generators to learn the shareable attribute from the other domain based on feature covariance matching. In this way, Resembled GAN handles semantic attributes such as color mood better than CoGAN. More importantly, Resembled GAN is more flexible to handle various levels of similarities between two domains. We expect that our feature matching idea can be extended toward cross-domain transfers in a unsupervised unconditional manner.

6 Acknowledgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative Program (IITP-2018-2017-0-01015) supervised by the IITP(Institute for Information & communications Technology Promotion), the Ministry of Science and ICT, Korea (2018-0-00207, Immersive Media Research Laboratory), the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the MSIP (NRF-2016R1A2B4016236), and ICT R&D program of MSIP/IITP. [R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding]

References

[1] Duhyeon Bang and Hyunjung Shim. Mggan: Solving mode collapse using manifold guided training. *arXiv preprint arXiv:1804.04391*, 2018.

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

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

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

[5] 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.

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[9] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[10] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[11] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

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

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

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

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

[16] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017.

[18] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 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.

[19] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint*, 2017.

[20] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for deep face recognition with long-tail data. *arXiv preprint arXiv:1803.09014*, 2018.

[21] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pages 802–816. Springer, 2008.

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

[23] 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*, pages 465–476, 2017.