What Would Cats Look Like as Anime Girls? Unsupervised Prototypical GAN for Image-to-Image Translation

Nandi Zhang
HKUST
nzhangag@connect.ust.hk

Figure 1. Samples drawn from *Animal2Anime* dataset for reference. Images in each domain are unlabeled and unpaired.

Abstract

Despite the recent success of cycle-consistent Generative Adversarial Networks (GANs) in unsupervised image-to-image translation, translation between heterogeneous domains under the cycle-consistency assumption remains challenging. Previous methods mostly focus on the translation between domains that are similar either in style (e.g., cat2dog, zebra2horse) or shape (e.g., photo2vangogh, selfie2anime). These methods usually fall short to handle concurrent translations in both style and shape across heterogeneous domains that share very few common representations. In this work, we propose a new approach called prototypical GAN that adapts to large domains differences under the cycle-consistency constraints. Prototypical GAN incorporates a siamese network that learns a feature vector based on an estimated domain prototype for each domain. The siamese network then guides each generator to generate translated images with similar feature vectors to the feature vectors of the original images. Feature vectors based on domain prototypes serve as higher-level features that would not hinder the transformation as low-level features do when the source and target domains are drastically different. We also build a heterogeneous-domain image-to-image translation dataset named Animal2Anime where prototypical GAN achieves impressive performance.

1. Introduction

Image-to-image translation between source and target domains has a broad range of applications such as colorizaiton [37], super resolution [21, 35], style transfer [10, 14] and image inpainting [15, 29], *etc*. The translation between domains using GANs [11] is essentially to learn a mapping between source and target domain distributions. Since labeled and well-paired datasets are often expensive to obtained and limited in scale, many unsupervised methods on unpaired image-to-image translation have been proposed in recent years [9, 16, 19, 26, 30, 33, 39, 41].

Despite the effectiveness of GANs in a large variety of image-to-image translation tasks, GAN-based methods mostly transform images from one domain to another by transferring local features (*e.g.*, textures, colors, contours, *etc*) [22,25,33,42]. However, local feature transfer becomes impractical in heterogeneous translation. On the one hand, transformations of low-level features between domains with large gaps are perplexed and lack of generality. On the other hand, previous methods usually impose certain constraints on the translation to preserve as much content representa-

tions in the translated results as possible [10, 19, 29, 30, 33]. This preservation potentially impede thorough transformations of low-level features in heterogeneous scenarios where well-translated images are expected to maintain very few original local features. In this paper, we look into the problem of image-to-image translation between heterogeneous domains. We argue that domain-specific features are crucial to heterogeneous translation, while previous methods mostly utilize domain-agnostic features.

Humans keep in mind a general representation of a certain domain, and features humans understand are in fact domain-specific. For example, when we consider what a gray cat would look like as a human being, apparently it will not be a gray-skin human. Since gray fur in cats is not an equivalent feature to gray skin in humans. However, the use of domain-agnostic features often leads to such undesirable mapping and feature invariant in heterogeneous translation. Therefore, we seek to let the model "keep in mind" a common representation for each domain, which is the domain prototype. With the domain prototype, we can derive a feature vector for each image that represents its domain-specific feature. A domain-specific feature vector is the departure of a encoded result from the domain prototype, and each entry of the feature vector intuitively indicates how a particular feature of the image is deviated from the common representation. Our experiment show that image-to-image translation with the preservation of domain-specific features produce promising results in our Animal2Anime dataset (Figure 1). In this work, our contribution can be summarized as follows:

- We introduce a novel prototypical GAN that tackles significant domain differences in image-to-image translation. Prototypical GAN incorporates a siamese network to learn the high-level domain-specific feature vectors and encourage the generators to align these feature vectors between samples and translated images.
- We construct a dataset as a challenging evaluation benchmark of image-to-image translation between unpaired heterogeneous domains called Animal2Anime.
- We conduct experiments to demonstrate the effectiveness of our prototypical GAN on heterogeneous translation tasks.

2. Related works

Generative Adversarial Networks. Generative Adversarial Networks (GANs) [11] training is essentially striking a Nash equilibrium between generators and discriminators [13], which partially accounts for its great difficulty of convergence. During training, a generator aims to generate realistic images to fool a discriminator while the discriminator tries to distinguish generated images from real images [11]. Despite the training difficulty, GAN has achieved remarkable performances across a wide variety of tasks such as image generation [2, 4, 17, 38], image-to-image translation [9, 16, 19, 26, 30, 33, 39, 41], image inpainting [15, 29], image denoising [6, 18], super resolution [21, 35], semantic image manipulation [23, 34], *etc*. In this work, the GAN architecture from U-GAT-IT [19] is used to translate images between the source and target domains.

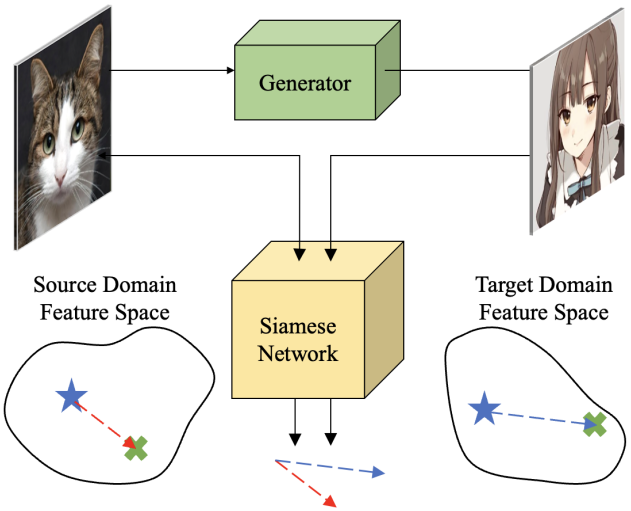

Figure 2. **Siamese Network –** The input and output images of the generator are fed into the siamese network to produce corresponding domain-specific feature vectors. The blue stars denote the domain prototypes of source and target domains. The green crosses denote where the images are encoded in each domain. The red vector denotes the feature vector of the sample image; the blue vector denotes the feature vector of the translated result of the sample image.

Image-to-Image Translation. In paired image-to-image translation, most works follow the conditional GAN framework proposed by [16]. Recently, more attention has been drawn to unpaired side of image-to-image translation [9, 19, 26, 30, 33, 41]. Without the ground truth data, cues for feature translations largely rely on common or distinct domain-agnostic features between the source and target domains. For example, CoGAN [27] constructs the domain mapping through a weight-sharing strategy that learns common representation across domains. Other methods [5, 31, 32] encourage the translated image to share common feature content with the input. AttentionGAN [33] identifies and masks the discriminative foregrounds between the source and target domains and minimizes the changes in the common background. U-GAT-IT [19] designed an auxiliary

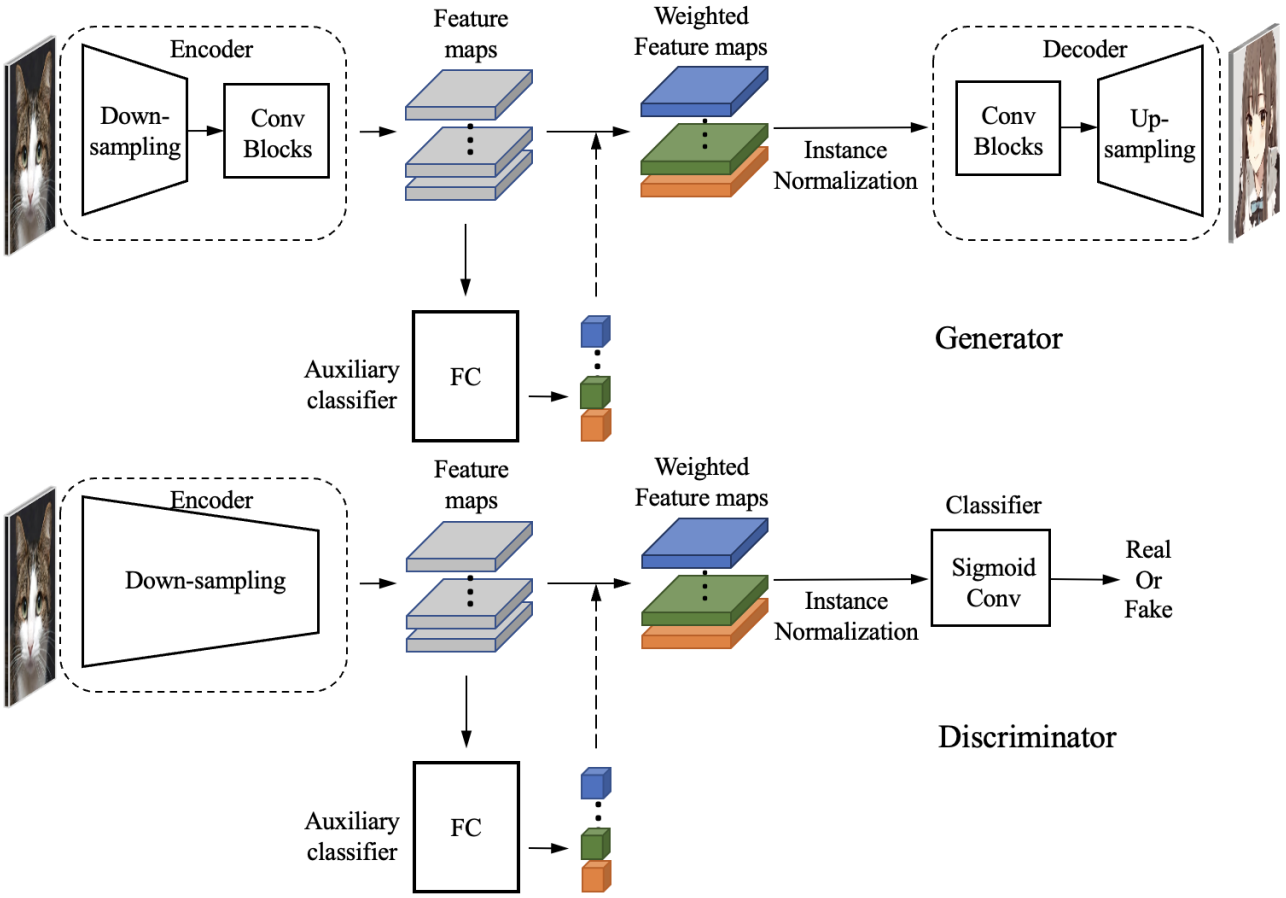

Figure 3. **Generators and Discriminators**

classifier to impose attention among the Class Activation Maps (CAM) [40] of feature maps implicitly. These methods are often limited to homogeneous translation. However, for domains that share almost no local common feature content no matter in shape or style, forcing the model to learn common representations between source and target domains would greatly hinder the transformation. So translation across very dissimilar domain may not allow us to follow these paradigms.

For heterogeneous translation, TransGaGa [36] uses a point-heatmap to estimate the geometry of source domain images and transfers it explicitly as a high-level semantic. TraVeLGAN [1] introduced a siamese network to learn the vector transformation in intra-domain latent space. It preserves the vector transformation of two arbitrary samples while eliminating cycle consistency constraints [41]. Our method is built upon the architecture of U-GAT-IT [19] but does not enforce preservation of low-level features. Though we share the idea of vector transformation with TraVeLGAN [1], our method may yield more stable and general results since we maintain the cycle consistency constraint and use domain prototypes which have a smaller variance than arbitrary samples.

3. Methodology

The task we investigate can be formulated as unpaired image-to-image translation between the animal face domain S and the anime face domain T. Our goal is to train a generator network $G_{S\to T}$ that maps image instances from the source domain X_S to the target domain X_T with only unpaired samples drawn from each of the two domains.

3.1. Model

Our models mainly follow the U-GAT-IT [19] framework which consists of two generators $G_{S\to T}$ and $G_{T\to S}$,and two discriminators D_S and D_T. For simplicity purposes, we only describe $G_{S\to T}$, D_T (Figure 3) and the siamese network in the forward cycle (Figure 2). The counter part is similar.

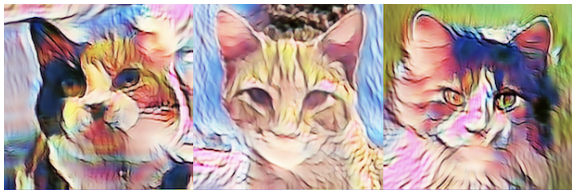

Figure 4. **Early Stage Results –** Source domain local features are largely preserved in the early iterations, while attention feature map mechanism speed up elimination of the hindrance caused by this preservation.

3.1.1 Generator

The generator $G_{S\to T}$ consists of a encoder E_S, a decoder G_T and an auxiliary classifier $\eta_{G_{S\to T}}$. Denote $s^{(S)} \in X_S$ as a sample drawn from the source domains S. $E_S(s^{(S)})$ is the activated feature maps output by the encoder E_S. Let $E_S^k(s^{(S)})$ denote the k-th feature map. The auxiliary classifier $\eta_{G_{S\to T}}$ learns weights w_s for each of the feature map (*e.g.*, w_s^k is the weight for the k-th feature map).

$$\begin{aligned} a_{G_{S\to T}}(s^{(S)}) &= w_s \cdot E_S(s^{(S)}) \\ &= \{w_s^k \cdot E_S^k(s^{(S)}) | 1 < k < N\}. \end{aligned} \tag{1}$$

In Equation 1, $a_{G_{S\to T}}(s^{(S)})$ denotes the reweighted feature maps. The intuition behind is similar to imposing attention on the feature maps, so we also call $a_{G_{S\to T}}(s^{(S)})$ attention feature maps. U-GAT-IT uses this attention feature maps to let $G_{S\to T}$ and D_T know which part to improve in homogeneous translation, however, we found them also useful in heterogeneous translation problem. Since preservation of source domain features would sometimes hinder the transformation, the reweighting mechanism may decide which part of the feature maps are not informative and allow the generator to overlook them. The attention feature maps speed up the convergence and we can observe in the result images that source domain local features may be eliminated earlier (*e.g.*, at first we often observe a heavy amount of source domain local features in the translated result) (Figure 4).

The instance normalization module we adopt follows the Adaptive LayerInstance Normalization (AdaLIN) proposed by U-GAT-IT whose parameters are learned during training by a fully connected layer from the attention feature maps. It adaptively select a ratio between Instance normalization (IN) [14] and Layer Normalization (LN) [3] and combines the advantages of both normalization.

3.1.2 Discriminator

The discriminator D_T largely follows the architecture of $G_{S\to T}$, consisting of an encoder E_{D_T}, a classifier C_{D_T} and an auxiliary classifier η_{D_T}. We train E_{D_T} and η_{D_T} along with the discrimination of C_{D_T}. Let $t^* \in G_{S\to T}(X_S)$ denote an image translated by the generator. The discriminator $D_T(t^*)$ can be rewritten as equation 2:

$$\begin{aligned} D_T(t^*) &= C_{D_T}(a_{D_T}(t^*)) \\ &= C_{D_T}(w_{t^*} \cdot E_{D_T}(t^*)). \end{aligned} \tag{2}$$

3.1.3 Siamese Network

Our siamese network (Figure 2) aims to produce domain-specific feature vectors based on domain prototypes. The derivation of feature vectors is a two-step pipeline.

First, we estimate the domain prototypes in a moving average manner:

$$\mathcal{E}(Prototype^{(S)})_\alpha = \delta \cdot \mathcal{E}(Prototype^{(S)})_{\alpha-1} + (1-\delta) \cdot s_\alpha^{(S)}. \tag{3}$$

In Equation 3, $\mathcal{E}(Prototype^{(S)})_\alpha$ denotes an estimation of the prototype for domain S at time step α, δ denotes the decay rate of the estimation, and $s_\alpha^{(S)}$ denotes the sample vector in domain S at time step α. The estimation is a $1 \times N$ vector after global average pooling [24] where N is the number of output feature maps of the siamese network. Sample vectors computed have the same dimensions as the estimation. The past estimation is replaced by the new sample computed at a rate of $(1-\delta)$.

Second, we compute the feature vector of the sample image:

$$\begin{aligned} f(s_\alpha^{(S)}) &= s_\alpha^{(S)} - \mathcal{E}(Prototype^{(S)})_\alpha \\ &= s_\alpha^{(S)} - (\delta \cdot \mathcal{E}(Prototype^{(S)})_{\alpha-1} + (1-\delta) \cdot s_\alpha^{(S)}) \\ &= \delta(s_\alpha^{(S)} - \mathcal{E}(Prototype^{(S)})_{\alpha-1}). \end{aligned} \tag{4}$$

In Equation 4, $f(s_\alpha^{(S)})$ is the feature vector of sample s in domain S at time step α. It can be easily calculated with the sample vector and prototype estimation of current time step.

In each training iteration, domain-specific feature vectors of the original and translated images are computed, while the siamese network is trained to minimize the gap between them.

3.2. Loss Function

In this section, we introduce the full objective for prototypical GAN to optimize, which can be broken down to four loss functions. Here we only explain the loss function formulation in the forward cycle.

Adversarial Loss. The adversarial loss from Least Squares GAN [28] instead of vanilla GAN is adopted to stabilize the

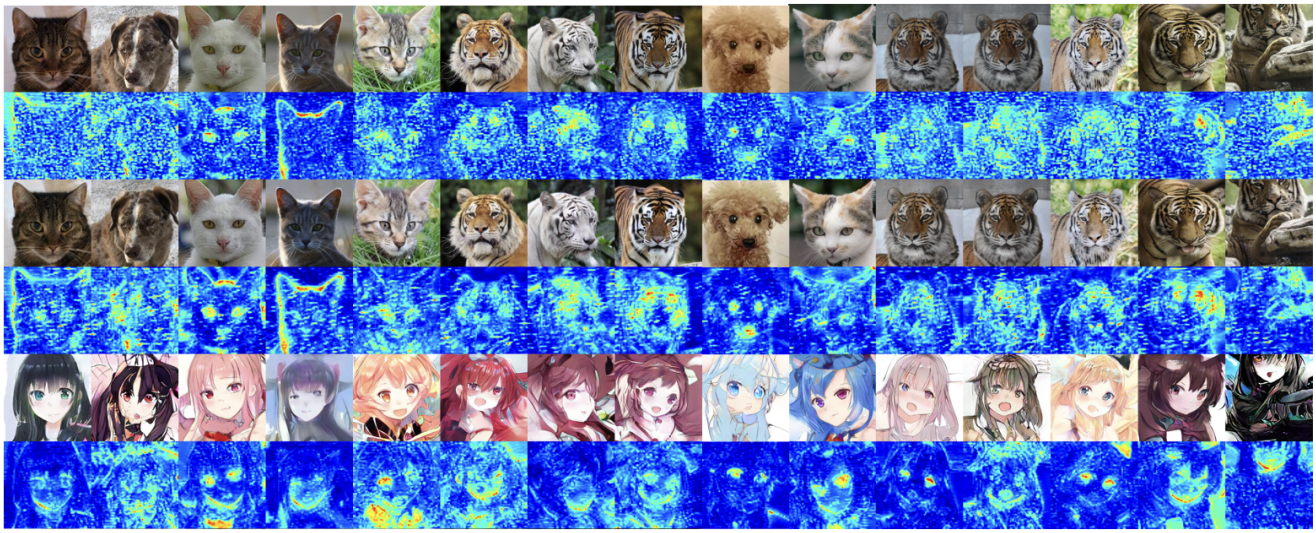

Figure 5. **Evaluation Results –** Test results of our trained prototypical GAN on Animal2Anime with CAM plots.

training:

$$L_{lsgan}^{S\to T} = \mathbf{E}_{x\sim X_T}[(D_T(x))^2] + \mathbf{E}_{x\sim X_S}[(1-D_T(G_{S\to T}(x)))^2]. \tag{5}$$

Cycle Loss. The cycle consistency constraint [41] helps alleviate the mode collapse problem, with an intuition that when we translate from one domain to another and back again, we should arrive at where we started.

$$L_{cycle}^{S\to T} = \mathbf{E}_{x\sim X_T}[|x - G_{T\to S}(G_{S\to T}(x))|_1]. \tag{6}$$

CAM Loss. We include the CAM loss from U-GAT-IT [19] to help the model identify which part of information should be use and which part should be overlook. Similar to U-GAT-IT, we use $\eta_{G_{S\to T}}(s^{(S)})$ to represent the probability that $s^{(S)}$ comes from X^S. Given an image $x \in \{X_S, X_T\}$, the CAM loss can be formulated as:

$$\begin{aligned} L_{cam}^{S\to T} = -(&\mathbf{E}_{x\sim X_S}[log(\eta_{G_{S\to T}}(x)] \\ &+\mathbf{E}_{x\sim X_T}[log(1-\eta_{G_{S\to T}}(x))]), \end{aligned} \tag{7}$$

$$L_{cam}^{D_T} = \mathbf{E}_{x\sim X_T}[(\eta_{D_T}(x))^2] + \mathbf{E}_{x\sim X_S}[(1-\eta_{D_T}(G_{S\to T}(x))^2]. \tag{8}$$

Prototypical Loss. We define the prototypical loss as the mean squared error between the feature vectors (See Equation 4 of $x \in X_S$ and its translated result $G_{S\to T}(x)$. We add a penalty term to prevent the model from always outputing 0 where ϵ represents the margin.

$$\begin{aligned} L_{prototype}^{S\to T} = &\|f(x) - f(G_{S\to T}(x))\|_2^2 \\ &+max(0, \epsilon - \|f(x)\|_2). \end{aligned} \tag{9}$$

This loss also serves as the loss function for the siamese network to optimize.

Full Objective. In the forward cycle, the generators and discriminators are jointly trained to optimize the full objective that combining loss functions from Equation 5, 6, 7, 9:

$$\begin{aligned} \mathbb{L}^{S\to T} = \min_{G_{S\to T}, \eta_{G_{S\to T}}} \max_{D_T, \eta_{D_T}} &\lambda_1 L_{lsgan}^{S\to T} + \lambda_2 L_{cycle}^{S\to T} \\ &+\lambda_3(L_{cam}^{S\to T} + L_{cam}^{D_T}) + \lambda_4 L_{prototype}^{S\to T} \end{aligned} \tag{10}$$

Here each λ represent the weight of each loss. In our implementation, we set $\lambda_1 = 1$, $\lambda_2 = 10$, $\lambda_3 = 1000$, $\lambda_4 = 10$. The final objective in the full cycle becomes:

$$\mathbb{L} = \mathbb{L}^{S\to T} + \mathbb{L}^{T\to S} \tag{11}$$

4. Experiments

We evaluate our model on the Animal2Anime dataset. In this section, we clarify some details and analyze on our experiment results.

4.1. Dataset Details

Our Animal2Anime dataset is comprised of an animal face dataset and a anime face dataset. The animal face dataset is Animal Faces-HQ (AFHQ) [7] including faces of cats, dogs and wildlife. The training dataset size is 16,130, and the test dataset size is 1,500, with resolution 512×512. The anime face dataset is the *another anime face dataset*[1]

[1]https://www.kaggle.com/scribbless/another-anime-face-dataset

Figure 6. **Translation Examples –** High-level features are preserved in the transformation. For example, the original and translated image of the tiger in the upper left corner are both partially visible and have a darker tone overall. Those of the tiger on the second row both have a tilted face. However, all translated images have very different local features from the original ones. Most of them form different hair and eye colors and the outlines of faces are well changed in an anime style.

crawled from Safebooru[2]. It contains around 92,200 high-quality anime faces with considerable variation in color, style, and posture, *etc*. We performed a random train-test split on the anime face dataset. The size of the training dataset is 90,000 and that of the test dataset is 2,200, with image resolution 256×256. All images in Animal2Anime are unpaired and unlabeled (Figure 1). All images will be resized to 256×256 during training and testing.

4.2. Attempts

Architecture. Though our GAN follow the framework of U-GAT-IT, the entire architecture turned out to be too large to train on a 16G GPU after added a siamese network. High-quality images seem very tempting in our dataset so we did not choose to sacrifice the image resolution eventually. So instead, we made some modifications to the model architecture. While following the overall structure of U-GAT-IT, we shrink the feature map channels of each part of the model by 75% to *64* (*e.g.*, the encoder down-sampling, the encoder bottleneck, the decoder bottleneck, and the auxiliary classifiers, *etc*.). And we discard the local discriminators in U-GAT-IT. Our siamese network adopts a backbone of ResNet-18 [12]. We use one as the batch size and Adam as the optimizer with $\beta_1 = 0.5, \beta_2 = 0.999$ [20]. The data augmentation and learning rate setting of U-GAT-IT also work for us.

[2]https://safebooru.org/

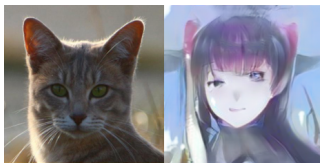

Figure 7. **Unsuccessful Transfer –** There are also some unsuccessful transfer result observed, most of which demonstrate an incomprehensive elimination of source domain local feature.

Algorithm. Before choosing a siamese network, we have attempted to use a fixed-parameter ResNet-50 pretrained on ImageNet [8] to derive feature vectors. However, the feature vectors of Anime dataset display a rather random pattern which is probably due to the lack of training data of similar style in ImageNet. On the contrary, ImageNet almost include all the animal types in our Animal dataset. So the enforced alignment of meaningless feature vectors with meaningful feature vectors is by no means helpful. Finally, we decide it necessary to train the feature vector derivation network on-the-fly with our GAN.

Hyperparameters. We mostly tuned the hyperparameters λ_1, λ_2, λ_3 and λ_4 in the full objective. Due to the computational complexity, we choose the range of each parameter based on their overall scaling and empirical experiments conducted by other practitioners. The hyperparameters may not be optimal though work fairly well so far.

4.3. Discussion on Evaluation Results

The evaluation results of prototypical GAN on Animal2Anime after 1,000,000 iterations of training is show in Figure 5. Interestingly, we observe that most transformed results share some high-level features (Figure 6) with the original images such as positions and posture (*i.e.*, the location and the rotation angle of the head). However, despite the use of attention feature maps, the transformation of source domain local features may not be thorough sometimes (Figure 7).

On the evaluation results, we have a few hypothesis for explanations and future improvement. First, since the image resolution we use is 256×256, and each feature map we encode them into has a size of 64×64 which is relatively large. So the decoder may not be able to completely get rid of the encoded source domain local features within a shallow stack of layers. The appropriate feature maps size may be much smaller for heterogeneous translation problem. Second, the weight of cycle loss might be too large, since the restriction of one-to-one mapping becomes exceptionally demanding due to the loose correspondence of heterogeneous domains.

5. Conclusions

In this work, we have proposed a prototypical GAN that tackles heterogeneous image-to-image translation problem via domain-specific features and an evaluation benchmark *Animal2Anime*. Our work explore the possibility of translating between two low-level-irrelevant but high-level-relevant domains without any manually selected features. Through experiments, we showed that our method yields promising results and analyzed some possible directions for improvement.

References

[1] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. *CoRR*, abs/1902.09631, 2019. 3

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 4

[4] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017. 2

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *CoRR*, abs/1612.05424, 2016. 2

[6] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018. 2

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[9] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference, 2017. 1, 2

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 1, 2

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 1, 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 6

[13] 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. *CoRR*, abs/1706.08500, 2017. 2

[14] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868, 2017. 1, 4

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), jul 2017. 1, 2

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 1, 2

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 2

[18] Dong-Wook Kim, Jae Ryun Chung, and Seung-Won Jung. Grdn:grouped residual dense network for real image denoising and gan-based real-world noise modeling, 2019. 2

[19] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *CoRR*, abs/1907.10830, 2019. 1, 2, 3, 5

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[21] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. 1, 2

[22] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. *CoRR*, abs/1808.00948, 2018. 1

[23] Xiaodan Liang, Hao Zhang, and Eric P. Xing. Generative semantic manipulation with contrasting gan, 2017. 2

[24] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network, 2014. 4

[25] Ming-Yu Liu, Thomas M. Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *CoRR*, abs/1703.00848, 2017. 1

[26] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *CoRR*, abs/1606.07536, 2016. 1, 2

[27] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *CoRR*, abs/1606.07536, 2016. 2

[28] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016. 4

[29] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016. 1, 2

[30] Paolo Russo, Fabio Maria Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive GAN. *CoRR*, abs/1705.08824, 2017. 1, 2

[31] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. *CoRR*, abs/1612.07828, 2016. 2

[32] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *CoRR*, abs/1611.02200, 2016. 2

[33] Hao Tang, Hong Liu, Dan Xu, Philip H. S. Torr, and Nicu Sebe. Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. *CoRR*, abs/1911.11897, 2019. 1, 2

[34] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans, 2018. 2

[35] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, abs/1809.00219, 2018. 1, 2

[36] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. *CoRR*, abs/1904.09571, 2019. 3

[37] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016. 1

[38] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016. 2

[39] Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, and Changyou Chen. Feature quantization improves GAN training. *CoRR*, abs/2004.02088, 2020. 1, 2

[40] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015. 3

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. 1, 2, 3, 5

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *CoRR*, abs/1711.11586, 2017. 1