U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation

Junho Kim[1], Minjae Kim[1], Hyeonwoo Kang[1], Kwanghee Lee[2*]
[1]NCSOFT
[2]Boeing Korea Engineering and Technology Center
{jhkimai, minjaekim, hwkang0131}@ncsoft.com
Kwanghee.lee2@boeing.com

Abstract

We propose a novel method for unsupervised image-to-image translation, which incorporates a new attention module and a new learnable normalization function in an end-to-end manner. The attention module guides our model to focus on more important regions distinguishing between source and target domains based on the attention map obtained by the auxiliary classifier. Unlike previous attention-based methods [29, 24] which cannot handle the geometric changes between domains, our model can translate both images requiring holistic changes and images requiring large shape changes. Moreover, our new AdaLIN (Adaptive Layer-Instance Normalization) function helps our attention-guided model to flexibly control the amount of change in shape and texture by learned parameters depending on datasets. Experimental results show the superiority of the proposed method compared to the existing state-of-the-art models with a fixed network architecture and hyper-parameters. Code is available at https://github.com/taki0112/UGATIT or https://github.com/znxlwm/UGATIT-pytorch.

1. Introduction

Image-to-image translation aims to learn a function that maps images within two different domains. This topic has gained a lot of attention from researchers in the fields of machine learning and computer vision because of its wide range of applications including image inpainting [32, 14], super resolution [7, 17], colorization [40, 41] and style transfer [9, 12]. When paired samples are given, the mapping model can be trained in a supervised manner using a conditional generative model [15, 22, 36] or a simple regression model [20, 26, 40]. In unsupervised settings where no paired data is available, multiple works [1, 6, 13, 18, 25, 33, 35, 39, 44] successfully have translated images using shared latent space [25] and cycle consistency assumptions [18, 44]. These works have been further developed to handle the multi-modality of the task [13].

Despite these advances, previous methods show performance differences depending on the amount of change in both shape and texture between domains. For example, they are successful for the style transfer tasks mapping local texture (*e.g.*, photo2vangogh and photo2portrait) but are typically unsuccessful for image translation tasks with larger shape change (*e.g.*, selfie2anime and cat2dog) in wild images. Therefore, the pre-processing steps such as image cropping and alignment are often required to avoid these problems by limiting the complexity of the data distributions [13, 25]. In addition, existing methods such as DRIT [21] cannot acquire the desired results for both image translation preserving the shape (*e.g.*, horse2zebra) and image translation changing the shape (*e.g.*, cat2dog) with the fixed network architecture and hyper-parameters. The network structure or hyper-parameter setting needs to be adjusted for the specific dataset.

In this work, we propose a novel method for unsupervised image-to-image translation, which incorporates a new attention module and a new learnable normalization function in an end-to-end manner. Previous attention-based works [29, 24] do not allow to transform the shape of the object because of attaching the background to the (translated) cropped objects. Unlike these works, Our model guides the translation to focus on more important regions and ignore minor regions by distinguishing between source and target domains based on the attention map obtained by the auxiliary classifier. These attention maps are embedded into the generator and discriminator to focus on semantically important areas, thus facilitating the shape transformation. While the attention map in the generator induces the focus on areas that specifically distinguish between the two domains, the attention map in the discriminator helps fine-tuning by

*corresponding author

focusing on the difference between real image and fake image in target domain.

In addition to the attentional mechanism, we have found that the choice of the normalization function has a significant impact on the quality of the transformed results for various datasets with different amounts of change in shape and texture. Inspired by Batch-Instance Normalization(BIN) [31], we propose Adaptive Layer-Instance Normalization (AdaLIN), whose parameters are learned from datasets during training time by adaptively selecting a proper ratio between Instance normalization (IN) and Layer Normalization (LN). The AdaLIN function helps our attention-guided model to flexibly control the amount of change in shape and texture. As a result, our model, without modifying the model architecture or the hyper-parameters, can perform image translation tasks not only requiring holistic changes but also requiring large shape changes. In the experiments, we show the superiority of the proposed method compared to the existing state-of-the-art models on not only style transfer but also object transfiguration. The main contribution of the proposed work can be summarized as follows:

- We propose a novel method for unsupervised image-to-image translation with a new attention module and a new normalization function, AdaLIN.
- Our attention module helps the model to know where to transform intensively by distinguishing between source and target domains based on the attention map obtained by the auxiliary classifier.
- AdaLIN function helps our attention-guided model to flexibly control the amount of change in shape and texture without modifying the model architecture or the hyper-parameters.

2. Related works

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GAN) [10] have achieved impressive results on a wide variety of image generation [2, 4, 16, 42], image inpainting [14], image translation [6, 13, 15, 25, 36, 44] tasks. In training, a generator aims to generate realistic images to fool a discriminator while the discriminator tries to distinguish the generated images from real images. Various multi-stage generative models [16, 36] and better training objectives [2, 4, 28, 42] have been proposed to generate more realistic images. In this paper, our model uses GAN to learn the transformation from a source domain to a significantly different target domain, given unpaired training data.

2.2. Image-to-image translation

Isola *et al*. [15] have proposed a conditional GAN-based unified framework for image-to-image translation. High-resolution version of the pix2pix have been proposed by Wang *et al*. [36] Recently, there have been various attempts [13, 18, 25, 35, 44] to learn image translation from an unpaired dataset. CycleGAN [44] have proposed a cyclic consistence loss for the first time to enforce one-to-one mapping. UNIT [25] assumed a shared-latent space to tackle unsupervised image translation. However, this approach performs well only when the two domains have similar patterns. MUNIT [13] makes it possible to extend to many-to-many mapping by decomposing the image into content code that is domain-invariant and a style code that captures domain-specific properties. MUNIT synthesizes the separated content and style to generate the final image, where the image quality is improved by using adaptive instance normalization [12]. With the same purpose as MUNIT, DRIT [21] decomposes images into content and style, so that many-to-many mapping is possible. The only difference is that content space is shared between the two domains using the weight sharing and content discriminator which is auxiliary classifier. Nevertheless, the performance of these methods [13, 25, 21] are limited to the dataset that contains well-aligned images between source and target domains.

2.3. Class Activation Map

Zhou *et al*. [43] have proposed Class Activation Map (CAM) using global average pooling in a CNN. The CAM for a particular class shows the discriminative image regions by the CNN to determine that class. In this work, our model leads to intensively change discriminative image regions provided by distinguishing two domains using the CAM approach. However, not only global average pooling is used, but global max pooling is also used to make the results better.

2.4. Normalization

Recent neural style transfer researches have shown that CNN feature statistics (*e.g*., Gram matrix [9], mean and variance [12]) can be used as direct descriptors for image styles. In particular, Instance Normalization (IN) has the effect of removing the style variation by directly normalizing the feature statistics of the image and is used more often than Batch Normalization (BN) or Layer Normalization (LN) in style transfer. However, when normalizing images, recent studies use Adaptive Instance Normalization (AdaIN) [12], Conditional Instance Normalization (CIN) [8], and Batch-Instance Normalization (BIN) [31] instead of using IN alone. In our work, we propose an Adaptive Layer-Instance Normalization (AdaLIN) function to adaptively select a proper ratio between IN and LN.

Through the AdaLIN, our attention-guided model can flexibly control the amount of change in shape and texture.

3. U-GAT-IT

Our goal is to train a function $G_{s\to t}$ that maps images from a source domain X_s to a target domain X_t using only unpaired samples drawn from each domain. Our framework consists of two generators $G_{s\to t}$ and $G_{t\to s}$ and two discriminators D_s and D_t. We integrate the attention module into both generator and discriminator. The attention module in the discriminator guides the generator to focus on regions that are critical to generate a realistic image. The attention module in the generator gives attention to the region distinguished from the other domain. Here, we only explain $G_{s\to t}$ and D_t (See Fig 1) as the vice versa should be straight-forward.

3.1. Model

3.1.1 Generator

Let $x \in \{X_s, X_t\}$ represent a sample from the source and the target domain. Our translation model $G_{s\to t}$ consists of an encoder E_s, a decoder G_t, and an auxiliary classifier η_s, where $\eta_s(x)$ represents the probability that x comes from X_s. Let $E_s^k(x)$ be the k-th activation map of the encoder and $E_s^{k_{ij}}(x)$ be the value at (i, j). Inspired by CAM [43], the auxiliary classifier is trained to learn the importance weights of the k-th feature map, w_s^k, by using the global average pooling and global max pooling, *i.e.*, $\eta_s(x) = \sigma(\Sigma_k w_s^k \Sigma_{ij} E_s^{k_{ij}}(x))$. By exploiting the importance weights, we can calculate a set of domain specific attention feature map $a_s(x) = w_s * E_s(x) = \{w_s^k E_s^k(x) | 1 \leq k \leq n\}$, where n is the number of encoded feature maps. Then, our translation model $G_{s\to t}$ becomes equal to $G_t(a_s(x))$. Inspired by recent works that use affine transformation parameters in normalization layers and combine normalization functions [12, 31], we equip the residual blocks with AdaLIN whose parameters are dynamically computed by a fully connected layer from the attention map.

$$\hat{a_I} = \frac{a - \mu_I}{\sqrt{\sigma_I^2 + \epsilon}}, \hat{a_L} = \frac{a - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}}, \tag{1}$$

$$AdaLIN(a, \gamma, \beta) = \gamma \cdot (\rho \cdot \hat{a_I} + (1 - \rho) \cdot \hat{a_L}) + \beta, \tag{2}$$

$$\rho \leftarrow clip_{[0,1]}(\rho - \tau \Delta\rho), \tag{3}$$

where μ_I, μ_L and σ_I, σ_L are channel-wise, layer-wise mean and standard deviation respectively, γ and β are parameters generated by the fully connected layer, τ is the learning rate and $\Delta\rho$ indicates the parameter update vector (*e.g.*, the gradient) determined by the optimizer. The values of ρ are constrained to the range of [0, 1] simply by imposing bounds at the parameter update step. Generator adjusts the value so that the value of ρ is close to 1 in the task where the instance normalization is important and the value of ρ is close to 0 in the task where the LN is important. The value of ρ is initialized to 1 in the residual blocks of the decoder and 0 in the up-sampling blocks of the decoder.

An optimal method to transfer the content features onto the style features is to apply Whitening and Coloring Transform (WCT) [23], but the computational cost is high due to the calculation of the covariance matrix and matrix inverse. Although, the AdaIN [12] is much faster than the WCT, it is sub-optimal to WCT as it assumes uncorrelation between feature channels. Thus the transferred features contain slightly more patterns of the content. On the other hand, the LN [3] does not assume uncorrelation between channels, but sometimes it does not keep the content structure of the original domain well because it considers global statistics only for the feature maps. To overcome this, our proposed normalization technique AdaLIN combines the advantages of AdaIN and LN by selectively keeping or changing the content information, which helps to solve a wide range of image-to-image translation problems.

3.1.2 Discriminator

Let $x \in \{X_t, G_{s\to t}(X_s)\}$ represent a sample from the target domain and the translated source domain. Similar to other translation models, the discriminator D_t consists of an encoder E_{D_t}, a classifier C_{D_t}, and an auxiliary classifier η_{D_t}. Unlike the other translation models, both $\eta_{D_t}(x)$ and $D_t(x)$ are now trained to discriminate whether x comes from X_t or $G_{s\to t}(X_s)$. Given a sample x, $D_t(x)$ exploits the attention feature maps $a_{D_t}(x) = w_{D_t} * E_{D_t}(x)$ using the importance weight w_{D_t} on the encoded feature maps $E_{D_t}(x)$ that is trained by $\eta_{D_t}(x)$. Then, our discriminator $D_t(x)$ becomes equal to $C_{D_t}(a_{D_t}(x))$.

3.2. Loss function

The full objective of our model comprises four loss functions. Here, instead of using the vanilla GAN objective, we used the Least Squares GAN [27] objective for stable training.

Adversarial loss An adversarial loss is employed to match the distribution of the translated images to the target image distribution:

$$\begin{aligned} L_{gan}^{s\to t} = &(\mathbb{E}_{x\sim X_t}[(D_t(x))^2] \\ &+ E_{x\sim X_s}[(1 - D_t(G_{s\to t}(x)))^2]) \end{aligned} \tag{4}$$

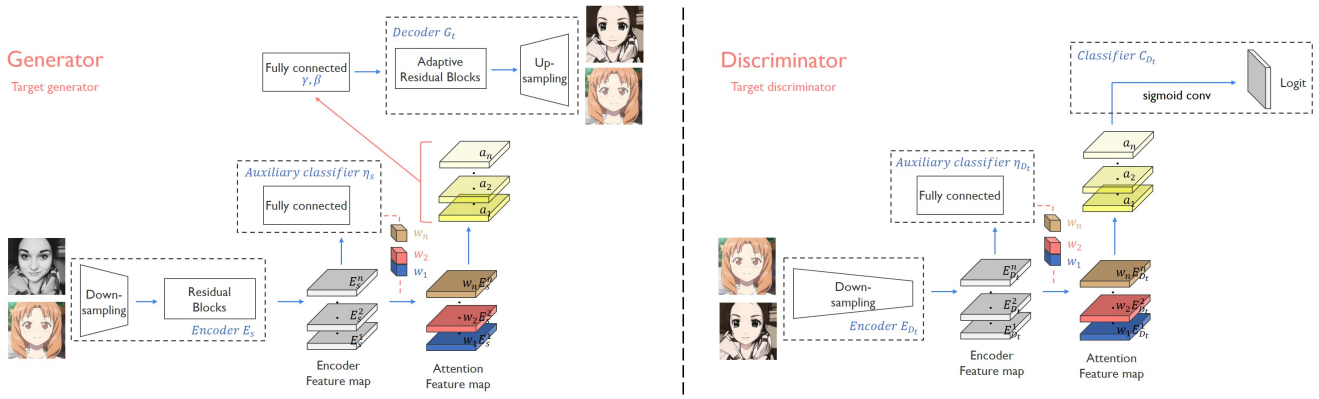


Figure 1. The model architecture of U-GAT-IT. The detailed notations are described in Section 3.1.

Cycle loss To alleviate the mode collapse problem, we apply a cycle consistency constraint to the generator. Given an image $x \in X_s$, after the sequential translations of x from X_s to X_t and from X_t to X_s, the image should be successfully translated back to the original domain:

$$L_{cycle}^{s \to t} = \mathbb{E}_{x \sim X_s}[|x - G_{t \to s}(G_{s \to t}(x))|_1] \tag{5}$$

Identity loss To ensure that the color distributions of input image and output image are similar, we apply an identity consistency constraint to the generator. Given an image $x \in X_t$, after the translation of x using $G_{s \to t}$, the image should not change.

$$L_{identity}^{s \to t} = \mathbb{E}_{x \sim X_t}[|x - G_{s \to t}(x)|_1] \tag{6}$$

CAM loss By exploiting the information from the auxiliary classifiers η_s and η_{D_t}, given an image $x \in \{X_s, X_t\}$. $G_{s \to t}$ and D_t get to know where they need to improve or what makes the most difference between two domains in the current state:

$$\begin{aligned} L_{cam}^{s \to t} = &-(\mathbb{E}_{x \sim X_s}[log(\eta_s(x))] \\ &+ \mathbb{E}_{x \sim X_t}[log(1 - \eta_s(x))], \end{aligned} \tag{7}$$

$$\begin{aligned} L_{cam}^{D_t} = &\mathbb{E}_{x \sim X_t}[(\eta_{D_t}(x))^2] \\ &+ \mathbb{E}_{x \sim X_s}[log(1 - \eta_{D_t}(G_{s \to t}(x)))^2] \end{aligned} \tag{8}$$

Full objective Finally, we jointly train the encoders, decoders, discriminators, and auxiliary classifiers to optimize the final objective:

$$\begin{aligned} \min_{G_{s \to t}, G_{t \to s}, \eta_s, \eta_t} \max_{D_s, D_t, \eta_{D_s}, \eta_{D_t}} &\lambda_1 L_{gan} + \\ &\lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam}, \end{aligned} \tag{9}$$

where $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 10, \lambda_4 = 1000$. Here, $L_{gan} = L_{gan}^{s \to t} + L_{gan}^{t \to s}$ and the other losses are defined in the similar way (L_{cycle}, $L_{identity}$, and L_{cam})

4. Implementation

4.1. Network architecture

The encoder of the generator is composed of two convolution layers with the stride size of two for down-sampling and four residual blocks. The decoder of the generator consists of four residual blocks and two up-sampling convolution layers with the stride size of one. Note that we use the instance normalization for the encoder and AdaLIN for the decoder, respectively. In general, LN does not perform better than batch normalization in classification problems [37]. Since the auxiliary classifier is connected from the encoder in the generator, to increase the accuracy of the auxiliary classifier we use the instance normalization(batch normalization with a mini-batch size of 1) instead of the AdaLIN. Spectral normalization [30] is used for the discriminator. We employ two different scales of PatchGAN [15] for the discriminator network, which classifies whether local (70 x 70) and global (286 x 286) image patches are real or fake. For the activation function, we use ReLU in the generator and leaky-ReLU with a slope of 0.2 in the discriminator.

4.2. Training

All models are trained using Adam [19] with β_1=0.5 and β_2=0.999. For data augmentation, we flipped the images horizontally with a probability of 0.5, resized them to 286 x 286, and random cropped them to 256 x 256. The batch size is set to one for all experiments. We train all models with a fixed learning rate of 0.0001 until 500,000 iterations and linearly decayed up to 1,000,000 iterations. We also use a weight decay at rate of 0.0001. The weights are initialized from a zero-centered normal distribution with a standard deviation of 0.02.

5. Experiments

5.1. Baseline model

We have compared our method with various models including CycleGAN [44], UNIT [25], MUNIT [13], and DRIT [21]. All the baseline methods are implemented using the author's code.

CycleGAN uses an adversarial loss to learn the mapping between two different domains X and Y. This method regularizes the mapping via cycle consistency losses. It uses two down-sampling convolution blocks, nine residual blocks, two up-sampling deconvolution blocks and four discriminator layers.

UNIT consists of two VAE-GAN with shared latent space. The structure of UNIT is similar to CycleGAN, but UNIT is different from CycleGAN in that it uses a multi-scale discriminator and shares the weight of the high-level layer stage of the encoder and decoder.

MUNIT can generate various outputs for a single input image. MUNIT assumes that the image representation can be decomposed into a content code and a style code. The difference between MUNIT's network structure and other networks is that MUNIT uses AdaIN in the decoder and a multi-scale discriminator.

DRIT is the most recent work associated with unsupervised image-to-image translation. DRIT can also generate various outputs for a single input image like MUNIT. Similar to MUNIT, it decomposes the image into a content code and a style code and uses a multi-scale discriminator. The difference is that the content code is shared like UNIT.

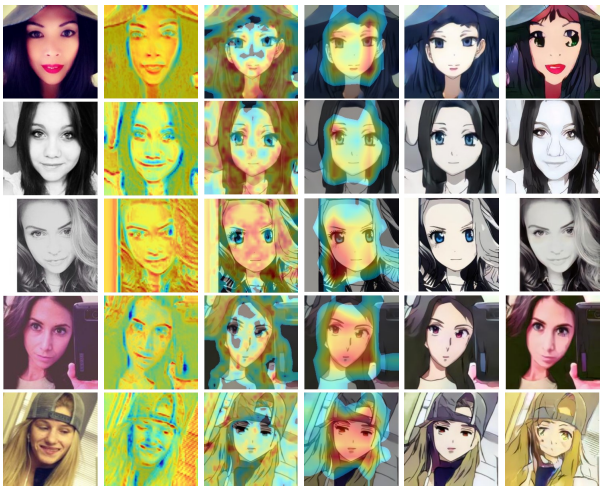
5.2. Dataset

We have evaluated the performance of each method with five unpaired image datasets including four representative image translation datasets and a newly created dataset consisting of real photos and animation artworks, *i.e.*, selfie2anime. All images are resized to 256 x 256 for training.

selfie2anime The selfie dataset contains 46,836 selfie images annotated with 36 different attributes. However, we only use photos of females as training data and test data. The size of the training dataset is 3400, and that of the test dataset is 100, with the image size of 256 x 256. For the anime dataset, we have firstly retrieved 69,926 animation character images from Anime-Planet[1]. Among those images, 27,023 face images are extracted by using an anime-face detector[2]. After selecting only female character images and removing monochrome images manually, we have collected two datasets of female anime face images, with the sizes of 3400 and 100 for training and test data respectively, which is the same numbers as the selfie dataset. Finally, all anime face images are resized to 256 x 256 by applying a CNN-based image super-resolution algorithm[3].

[1] http://www.anime-planet.com/

[2] https://github.com/nagadomi/lbpcascade_animeface



(a) (b) (c) (d) (e) (f)

Figure 2. Visualization of the attention maps and their effects shown in the ablation experiments: (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminator, respectively. (e) Our results with CAM, (f) Results without CAM.

horse2zebra and photo2vangogh These datasets are used in CycleGAN [44]. The training dataset size of each class: 1,067 (horse), 1,334 (zebra), 6,287 (photo), and 400 (vangogh). The test datasets consist of 120 (horse), 140 (zebra), 751 (photo), and 400 (vangogh). Note that the training data and the test data of vangogh class are the same.

cat2dog and photo2portrait These datasets are used in DRIT [21]. The numbers of data for each class are 871 (cat), 1,364 (zebra), 6,452 (photo), and 1,811 (vangogh). We use 120 (horse), 140 (zebra), 751 (photo), and 400 (vangogh) randomly selected images as test data, respectively.

5.3. Experiment results

We first analyze the effects of attention module and AdaLIN in the proposed model. We then compare the performance of our model against the other unsupervised image translation models listed in the previous section. To evaluate, the visual quality of translated images, we have conducted a user study. Users are asked to select the best image among the images generated from five different methods. More examples of the results from our model are included in the supplementary materials.

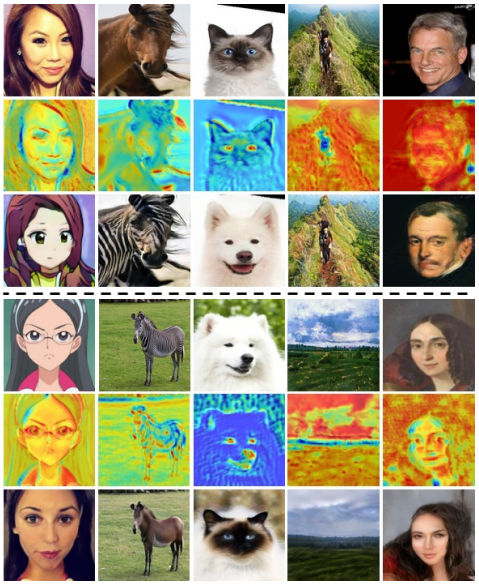


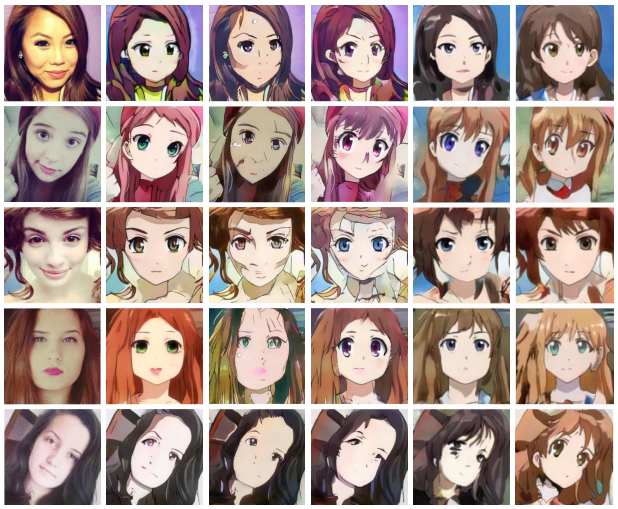
Figure 3. Visualization of the attention maps for various datasets: The first and fourth rows are the source images, the second and fifth rows are the attention map of generator, the third and sixth rows are our results.

5.3.1 CAM analysis

First, we conduct an ablation study to confirm the benefit from the attention modules used in both generator and discriminator. As shown in Fig 2 (b), the attention feature map helps the generator to focus on the source image regions that are more discriminative from the target domain, such as eyes and mouth. Meanwhile, we can see the regions where the discriminator concentrates its attention to determine whether the target image is real or fake by visualizing local and global attention maps of the discriminator as shown in Fig 2 (c) and (d), respectively. The generator can fine-tune the area where the discriminator focuses on with those attention maps. Note that we incorporate both global and local attention maps from two discriminators having different size of receptive field. Those maps can help the generator to capture the global structure (*e.g.*, face area and near of eyes) as well as the local regions. With this information some regions are translated with more care. The results with the attention module shown in Fig 2 (e) verify the advantageous effect of exploiting attention feature map in an image translation task. On the other hand, one can see that the eyes are misaligned, or the translation is not done at all in the results without using attention module as shown in Fig 2 (f). The attention map for the other datasets is shown in Fig 3.

[3]https://github.com/nagadomi/waifu2x

5.3.2 AdaLIN analysis



(a) (b) (c) (d) (e) (f)

Figure 4. Comparison of the results using each normalization function: (a) Source images, (b) Our results, (c) Results only using IN in decoder with CAM, (d) Results only using LN in decoder with CAM, (e) Results only using AdaIN in decoder with CAM, (f) Results only using GN in decoder with CAM.

As described in Sec 4.1, we have applied the AdaLIN only to the decoder of the generator. The role of the residual blocks in the decoder is to embed features, and the role of the up-sampling convolution blocks in the decoder is to generate target domain images from the embedded features. If the learned value of the gate parameter ρ is closer to 1, it means that the corresponding layers rely more on IN than LN. Likewise, if the learned value of ρ is closer to 0, it means that the corresponding layers rely more on LN than IN. As shown in Fig 4 (c), in the case of using only IN in the decoder, the features of the source domain (*e.g.*, earrings and shades around cheekbones) are well preserved due to channel-wise normalized feature statistics used in the residual blocks. However, the amount of translation to target domain style is somewhat insufficient since the global style cannot be captured by IN of the up-sampling convolution blocks. On the other hand, As shown in Fig 4 (d), if we use only LN in the decoder, target domain style can be transferred sufficiently by virtue of layer-wise normalized feature statistics used in the up-sampling convolution. But the features of the source domain image are less preserved by using LN in the residual blocks. This analysis of two extreme cases tells us that it is beneficial to rely more on IN than LN in the feature representation layers to preserve semantic characteristics of source domain, and the opposite is true for the up-sampling layers that actually generate images from the feature embedding. Therefore, the proposed

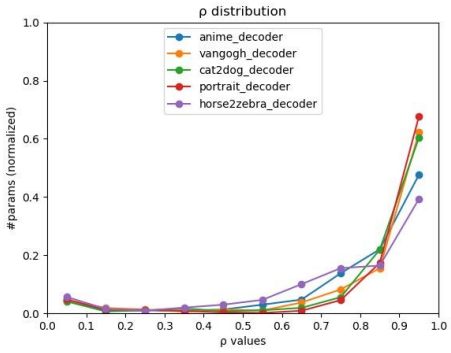


Figure 5. Distributions of the ρ in decoder for the various datasets

Table 1. Kernel Inception Distance$\times$100$\pm$std.$\times$100 for ablation our model. Lower is better. There are some notations; GN: Group Normalization, G_CAM: CAM of generator, D_CAM: CAM of discriminator

Model \ Dataset	selfie2anime	anime2selfie
U-GAT-IT	**11.61 $\pm$ 0.57**	**11.52 $\pm$ 0.57**
U-GAT-IT w/ IN	13.64 $\pm$ 0.76	13.58 $\pm$ 0.8
U-GAT-IT w/ LN	12.39 $\pm$ 0.61	13.17 $\pm$ 0.8
U-GAT-IT w/ AdaIN	12.29 $\pm$ 0.78	11.81 $\pm$ 0.77
U-GAT-IT w/ GN	12.76 $\pm$ 0.64	12.30 $\pm$ 0.77
U-GAT-IT w/o CAM	12.85 $\pm$ 0.82	14.06 $\pm$ 0.75
U-GAT-IT w/o G_CAM	12.33 $\pm$ 0.68	13.86 $\pm$ 0.75
U-GAT-IT w/o D_CAM	12.49 $\pm$ 0.74	13.33 $\pm$ 0.89

AdaLIN which adjusts the ratio of IN and LN in the decoder according to source and target domain distributions is more preferable in unsupervised image-to-image translation tasks. Additionally, the Fig 4 (e), (f) are the results of using the AdaIN and Group Normalization (GN) [38] respectively, and our methods are showing better results compared to these.

Also, as shwon in Table 1, we demonstrate the performance of the attention module and AdaLIN in the selfie2anime dataset through an ablation study using Kernel Inception Distance (KID) [5]. Our model achieves the lowest KID values. Even if the attention module and AdaLIN are used separately, we can see that our models perform better than the others. However, when used together, the performance is even better. As shown in Fig 5, the learned values of ρ vary depending on the dataset. Also, for the style transfer tasks like photo2vangogh, photo2portrait, the IN is known to perform well. The AdaLIN can also see that the ρ value is close to 1.

5.3.3 Qualitative evaluation

For qualitative evaluation, we have also conducted a perceptual study. 135 participants are shown translated results from different methods including the proposed method with source image, and asked to select the best translated image to target domain. We inform only the name of target domain, *i.e.*, animation, dog, and zebra to the participants. But, some example images of target domain are provided for the portrait and Van Gogh datasets as minimum information to ensure proper judgments. Table 2 shows that the proposed method achieved significantly higher score except for photo2vangogh but comparable in human perceptual study compared to other methods. In Fig 6 and 7, we present the image translation results from each method for performance comparisons. U-GAT-IT can generate undistorted image by focusing more on the distinct regions between source and target domain by exploiting the attention modules. Note that the regions around heads of two zebras or eyes of dog are distorted in the results from CycleGAN. Moreover, translated results using U-GAT-IT are visually superior to other methods while preserving semantic features of source domain as shown in the first and fourth rows of Fig 6 and 7. It is worth noting that the results from MUNIT and DRIT are much dissimilar to the source images since they generate images with random style codes for diversity. Furthermore, it should be emphasized that U-GAT-IT have applied with the same network architecture and hyper-parameters for all of the five different datasets, while the other algorithms are trained with preset networks or hyper-parameters. Through the results of user study, we show that the combination of our attention module and AdaLIN makes our model data-agnostic.

5.3.4 Quantitative evaluation

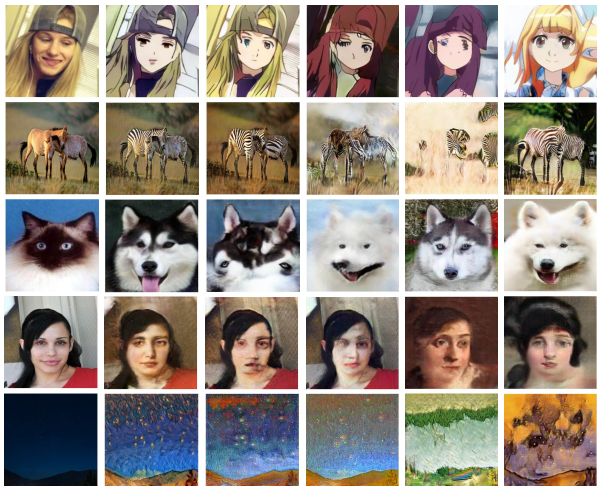
For quantitative evaluation, we use the recently proposed KID, which computes the squared Maximum Mean Discrepancy between the feature representations of real and generated images. The feature representations are extracted from the Inception network [34]. In contrast to the Frchet Inception Distance [11], KID has an unbiased estimator, which makes it more reliable, especially when there are fewer test images than the dimensionality of the inception features. The lower KID indicates that the more shared visual similarities between real and generated images [29]. Therefore, if well translated, the KID will have a small value in several datasets. Table 3 shows that the proposed method achieved the lowest KID scores except for the style transfer tasks like photo2vangogh and photo2portrait. However, there is no big difference from the lowest score. Also, unlike UNIT and MUNIT, we can see that the source $\rightarrow$ target, target $\rightarrow$ source translations are both stable.

Table 2. Preference score on translated images by user study.

Model \ Dataset	selfie2anime	horse2zebra	cat2dog	photo2portrait	photo2vangogh
U-GAT-IT	**73.15**	**73.56**	**58.22**	30.59	**48.96**
CycleGAN	20.07	23.07	6.19	26.59	27.33
UNIT	1.48	0.85	18.63	**32.11**	11.93
MUNIT	3.41	1.04	14.48	8.22	2.07
DRIT	1.89	1.48	2.48	2.48	9.70

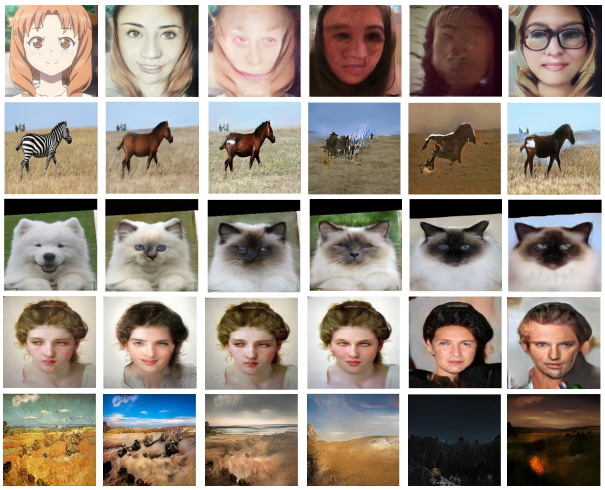
Table 3. Kernel Inception Distance×100±std.×100 for difference image translation mode. Lower is better.

Model \ Dataset	selfie2anime	horse2zebra	cat2dog	photo2portrait	photo2vangogh
U-GAT-IT	**11.61 ± 0.57**	**7.06 ± 0.8**	**7.07 ± 0.65**	1.79 ± 0.34	4.28 ± 0.33
CycleGAN	13.08 ± 0.49	8.05 ± 0.72	8.92 ± 0.69	1.84 ± 0.34	5.46 ± 0.33
UNIT	14.71 ± 0.59	10.44 ± 0.67	8.15 ± 0.48	**1.2 ± 0.31**	**4.26 ± 0.29**
MUNIT	13.85 ± 0.41	11.41 ± 0.83	10.13 ± 0.27	4.75 ± 0.52	13.08 ± 0.34
DRIT	15.08 ± 0.62	9.79 ± 0.62	10.92 ± 0.33	5.85 ± 0.54	12.65 ± 0.35
Model \ Dataset	anime2selfie	zebra2horse	dog2cat	portrait2photo	vangogh2photo
U-GAT-IT	**11.52 ± 0.57**	**7.47 ± 0.71**	**8.15 ± 0.66**	1.69 ± 0.53	5.61 ± 0.32
CycleGAN	11.84 ± 0.74	8.0 ± 0.66	9.94 ± 0.36	1.82 ± 0.36	**4.68 ± 0.36**
UNIT	26.32 ± 0.92	14.93 ± 0.75	9.81 ± 0.34	**1.42 ± 0.24**	9.72 ± 0.33
MUNIT	13.94 ± 0.72	16.47 ± 1.04	10.39 ± 0.25	3.3 ± 0.47	9.53 ± 0.35
DRIT	14.85 ± 0.6	10.98 ± 0.55	10.86 ± 0.24	4.76 ± 0.72	7.72 ± 0.34



(a) (b) (c) (d) (e) (f)

Figure 6. Visual comparisons on the five datasets. From top to bottom: selfie2anime, horse2zebra, cat2dog, photo2portrait, and photo2vangogh. (a) Source images, (b) U-GAT-IT, (c) CycleGAN, (d) UNIT, (e) MUNIT, (f) DRIT.



(a) (b) (c) (d) (e) (f)

Figure 7. Visual comparisons on the five datasets. From top to bottom: anime2selfie, zebra2horse, dog2cat, protrait2photo, and vangogh2photo. (a) Source images, (b) U-GAT-IT, (c) CycleGAN, (d) UNIT, (e) MUNIT, (f) DRIT.

6. Conclusions

In this paper, we have proposed unsupervised image-to-image translation (U-GAT-IT), with the attention mod-

ule and AdaLIN which can produce more visually pleasing results in various datasets with a fixed network architecture and hyper-parameter. Detailed analysis of various experimental results supports our assumption that attention maps obtained by an auxiliary classifier can guide generator to focus more on distinct regions between source and target domain. In addition, we have found that the Adaptive Layer-Instance Normalization (AdaLIN) is essential for translating various datasets that contains different amount of geometry and style changes. Through experiments, we have shown that the superiority of the proposed method compared to the existing state-of-the-art GAN-based models for unsupervised image-to-image translation tasks.

References

[1] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool. Combogan: Unrestrained scalability for image domain translation. *arXiv preprint arXiv:1712.06909*, 2017. 1

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017. 2

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[4] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 2

[5] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 7

[6] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint arXiv:1711.09020*, 2017. 1, 2

[7] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 1

[8] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2017. 2

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

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 7

[12] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR, abs/1703.06868*, 2:3, 2017. 1, 2, 3

[13] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018. 1, 2, 5, 12

[14] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 1, 2

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 1, 2, 4

[16] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[17] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 1

[18] T. Kim, M. Cha, H. Kim, J. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017. 1, 2

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[20] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016. 1

[21] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. *arXiv preprint arXiv:1808.00948*, pages 35–51, 2018. 1, 2, 5, 12

[22] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pages 5501–5509, 2017. 1

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017. 3

[24] X. Liang, H. Zhang, and E. P. Xing. Generative semantic manipulation with contrasting gan. *arXiv preprint arXiv:1708.00315*, 2017. 1

[25] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 1, 2, 5, 12

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[28] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017. 2

[29] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim. Unsupervised attention-guided image-to-image trans-

lation. In *Advances in Neural Information Processing Systems*, pages 3697–3707, 2018. 1, 7

[30] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 4

[31] H. Nam and H.-E. Kim. Batch-instance normalization for adaptively style-invariant neural networks. *arXiv preprint arXiv:1805.07925*, 2018. 2, 3

[32] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2014. 1

[33] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Moressi, F. Cole, and K. Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. *arXiv preprint arXiv:1711.05139*, 2017. 1

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 7

[35] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 1, 2

[36] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017. 1, 2

[37] Y. Wu and K. He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018. 4

[38] Y. Wu and K. He. Group normalization. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7

[39] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint*, 2017. 1

[40] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. 1

[41] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017. 1

[42] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[43] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016. 2, 3

[44] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 1, 2, 5, 12

A. Additional experimental results

In addition to the results presented in the paper, we show supplement generation results for the five datasets in Figs 8, 9, 10, 11, 12, 13, 14, and 15.

B. Network Architecture

The network architectures of U-GAT-IT are shown in Table 4, 5, and 6. For the generator network, we use instance normalization in encoders, adaptive layer-instance normalization in decoders, and no normalization in last output layer. For the discriminator, we use Leaky-ReLU with a negative slope of 0.2, spectral normalization in all layers. There are some notations; N: the number of output channels, K: kernel size, S: stride size, P: padding size, IN: instance normalization, AdaLIN: adaptive layer-instance normalization, LIN: layer-instance normalization, SN: spectral normalization.

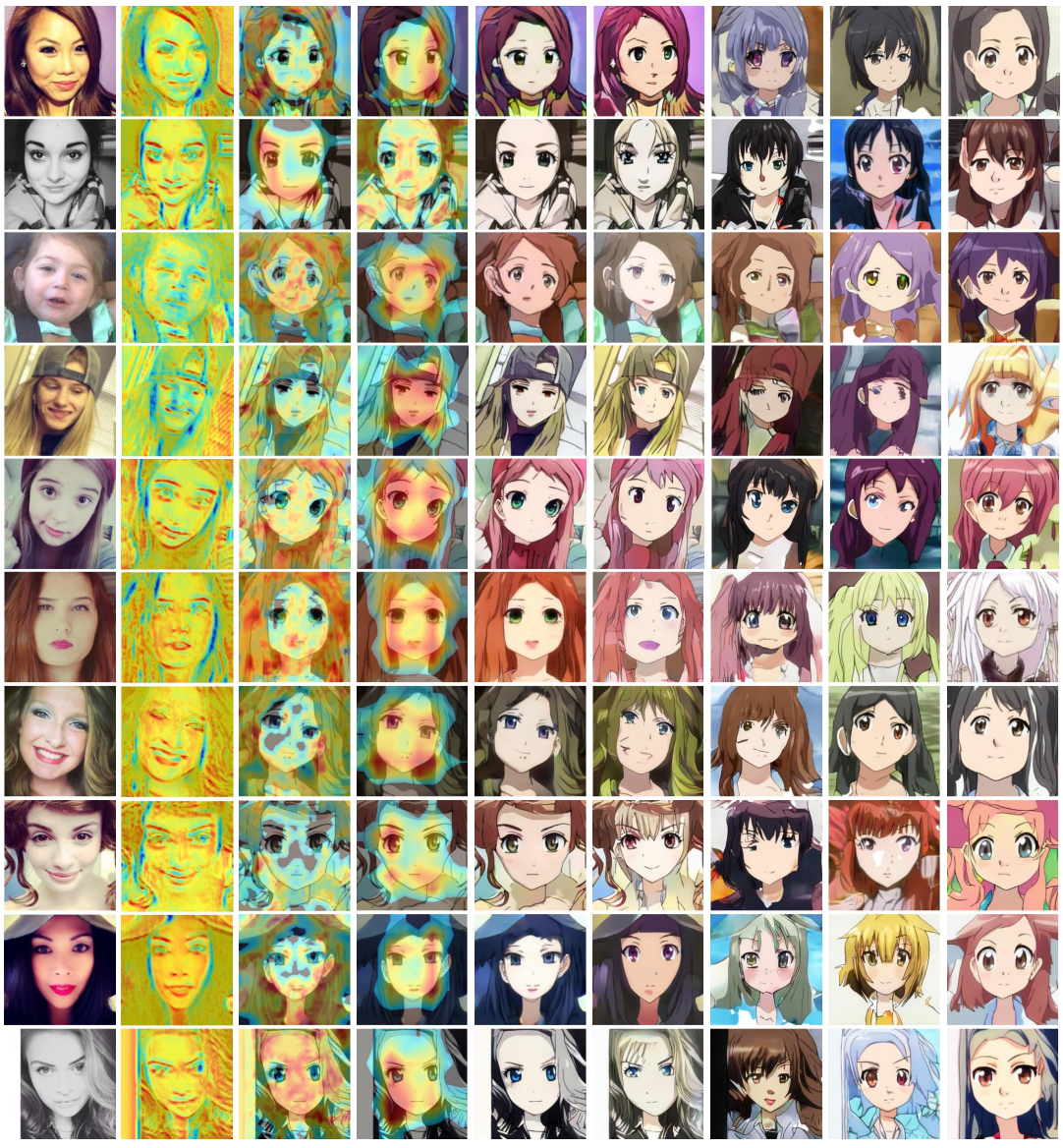


Figure 8. Visual comparisons of the selfie2anime with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN [44], (g) UNIT [25], (h) MUNIT [13], (i) DRIT [21].

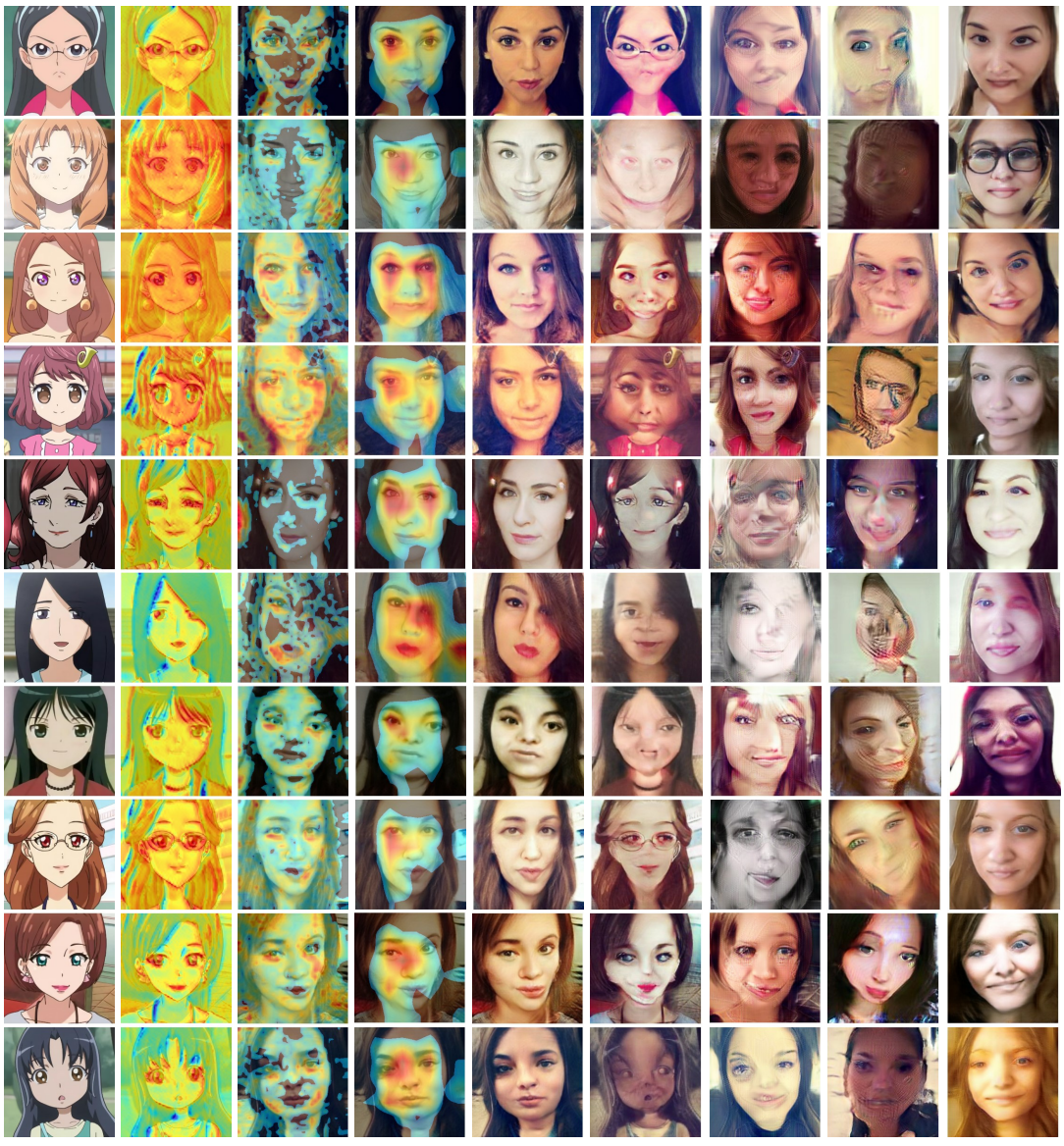
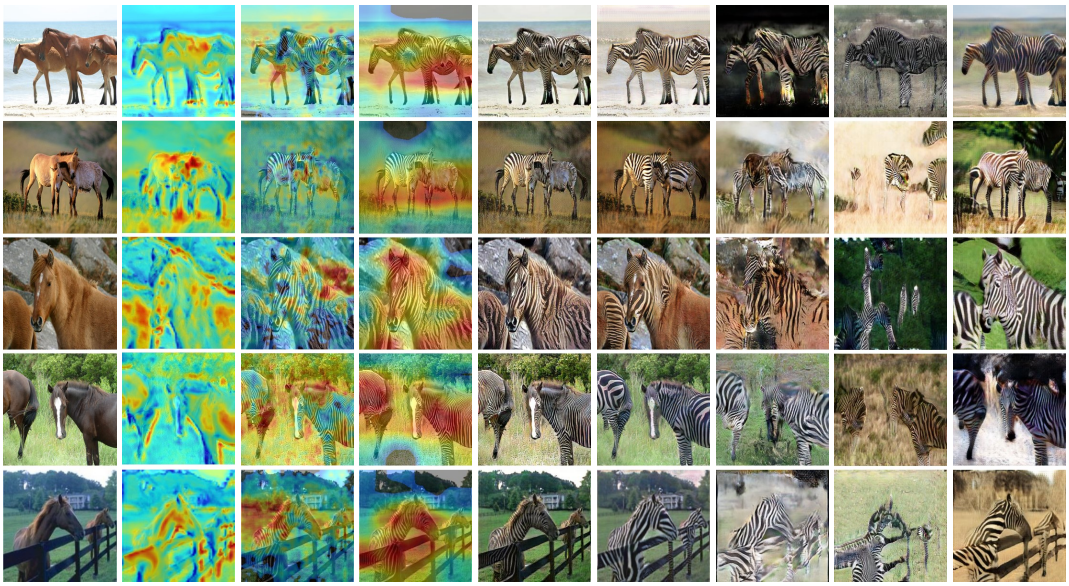


Figure 9. isual comparisons of the anime2selfie with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 10. Visual comparisons of the horse2zebra with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 11. Visual comparisons of the zebra2horse with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.

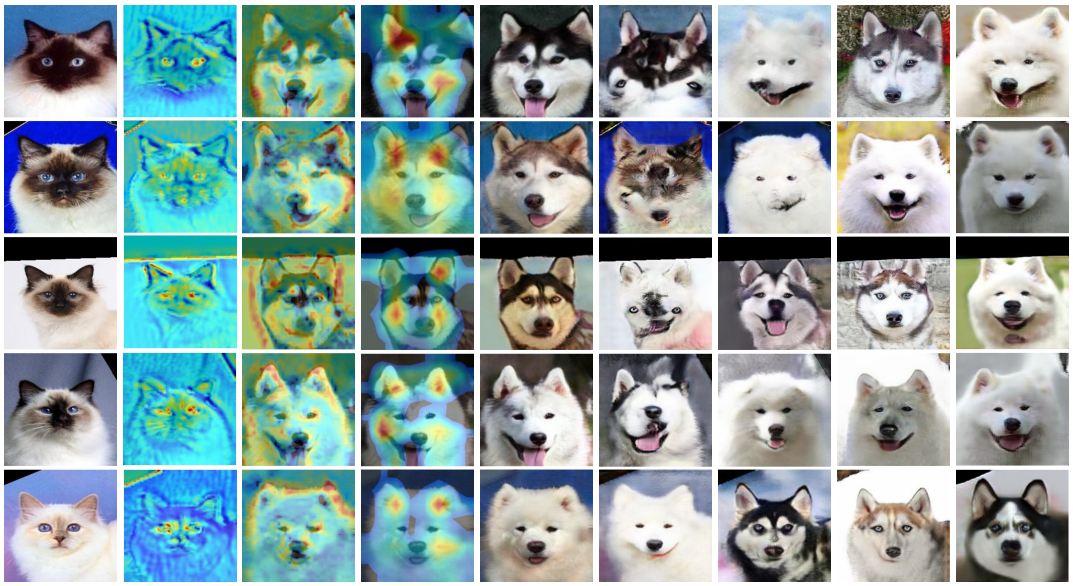


Figure 12. Visual comparisons of the cat2dog with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.

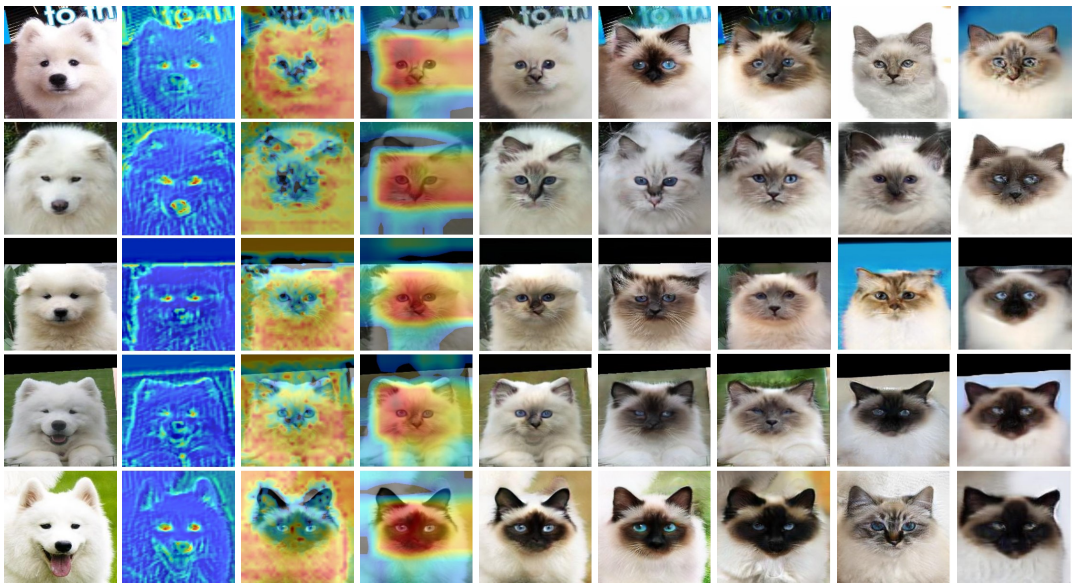
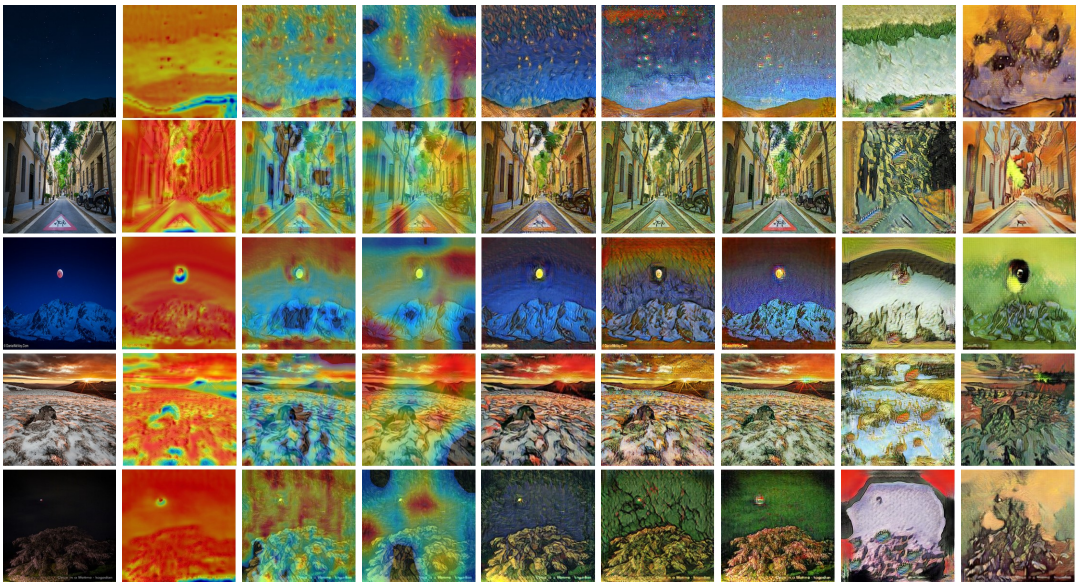


Figure 13. Visual comparisons of the dog2cat with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 14. Visual comparisons of the photo2vangogh with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, respectively, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 15. Visual comparisons of the photo2portrait with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, respectively, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT.

Table 4. The detail of generator architecture.

Part	Input → Output Shape	Layer Information
Encoder Down-sampling	(h, w, 3) → (h, w, 64)	CONV-(N64, K7, S1, P3), IN, ReLU
	(h, w, 64) → ($\frac{h}{2}$, $\frac{w}{2}$, 128)	CONV-(N128, K3, S2, P1), IN, ReLU
	($\frac{h}{2}$, $\frac{w}{2}$, 128) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	CONV-(N256, K3, S2, P1), IN, ReLU
Encoder Bottleneck	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	ResBlock-(N256, K3, S1, P1), IN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	ResBlock-(N256, K3, S1, P1), IN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	ResBlock-(N256, K3, S1, P1), IN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	ResBlock-(N256, K3, S1, P1), IN, ReLU
CAM of Generator	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 512)	Global Average & Max Pooling, MLP-(N1), Multiply the weights of MLP
	($\frac{h}{4}$, $\frac{w}{4}$, 512) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	CONV-(N256, K1, S1), ReLU
γ, β	($\frac{h}{4}$, $\frac{w}{4}$, 256) → (1, 1, 256)	MLP-(N256), ReLU
	(1, 1, 256) → (1, 1, 256)	MLP-(N256), ReLU
	(1, 1, 256) → (1, 1, 256)	MLP-(N256), ReLU
Decoder Bottleneck	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	AdaResBlock-(N256, K3, S1, P1), AdaILN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	AdaResBlock-(N256, K3, S1, P1), AdaILN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	AdaResBlock-(N256, K3, S1, P1), AdaILN, ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{4}$, $\frac{w}{4}$, 256)	AdaResBlock-(N256, K3, S1, P1), AdaILN, ReLU
Decoder Up-sampling	($\frac{h}{4}$, $\frac{w}{4}$, 256) → ($\frac{h}{2}$, $\frac{w}{2}$, 128)	Up-CONV-(N128, K3, S1, P1), LIN, ReLU
	($\frac{h}{2}$, $\frac{w}{2}$, 128) → (h, w, 64)	Up-CONV-(N64, K3, S1, P1), LIN, ReLU
	(h, w, 64) → (h, w, 3)	CONV-(N3, K7, S1, P3), Tanh

Table 5. The detail of local discriminator.

Part	Input → Output Shape	Layer Information
Encoder Down-sampling	(h, w, 3) → ($\frac{h}{2}$, $\frac{w}{2}$, 64)	CONV-(N64, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{2}$, $\frac{w}{2}$, 64) → ($\frac{h}{4}$, $\frac{w}{4}$, 128)	CONV-(N128, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 128) → ($\frac{h}{8}$, $\frac{w}{8}$, 256)	CONV-(N256, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{8}$, $\frac{w}{8}$, 256) → ($\frac{h}{8}$, $\frac{w}{8}$, 512)	CONV-(N512, K4, S1, P1), SN, Leaky-ReLU
CAM of Discriminator	($\frac{h}{4}$, $\frac{w}{4}$, 512) → ($\frac{h}{4}$, $\frac{w}{4}$, 1024)	Global Average & Max Pooling, MLP-(N1), Multiply the weights of MLP
	($\frac{h}{4}$, $\frac{w}{4}$, 1024) → ($\frac{h}{4}$, $\frac{w}{4}$, 512)	CONV-(N512, K1, S1), Leaky-ReLU
Classifier	($\frac{h}{4}$, $\frac{w}{4}$, 512) → ($\frac{h}{4}$, $\frac{w}{4}$, 1)	CONV-(N1, K4, S1, P1), SN

Table 6. The detail of global discriminator.

Part	Input → Output Shape	Layer Information
Encoder Down-sampling	(h, w, 3) → ($\frac{h}{2}$, $\frac{w}{2}$, 64)	CONV-(N64, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{2}$, $\frac{w}{2}$, 64) → ($\frac{h}{4}$, $\frac{w}{4}$, 128)	CONV-(N128, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{4}$, $\frac{w}{4}$, 128) → ($\frac{h}{8}$, $\frac{w}{8}$, 256)	CONV-(N256, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{8}$, $\frac{w}{8}$, 256) → ($\frac{h}{16}$, $\frac{w}{16}$, 512)	CONV-(N512, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{16}$, $\frac{w}{16}$, 512) → ($\frac{h}{32}$, $\frac{w}{32}$, 1024)	CONV-(N1024, K4, S2, P1), SN, Leaky-ReLU
	($\frac{h}{32}$, $\frac{w}{32}$, 1024) → ($\frac{h}{32}$, $\frac{w}{32}$, 2048)	CONV-(N2048, K4, S1, P1), SN, Leaky-ReLU
CAM of Discriminator	($\frac{h}{32}$, $\frac{w}{32}$, 2048) → ($\frac{h}{32}$, $\frac{w}{32}$, 4096)	Global Average & Max Pooling, MLP-(N1), Multiply the weights of MLP
	($\frac{h}{32}$, $\frac{w}{32}$, 4096) → ($\frac{h}{32}$, $\frac{w}{32}$, 2048)	CONV-(N2048, K1, S1), Leaky-ReLU
Classifier	($\frac{h}{32}$, $\frac{w}{32}$, 2048) → ($\frac{h}{32}$, $\frac{w}{32}$, 1)	CONV-(N1, K4, S1, P1), SN