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

Anonymous authors
Paper under double-blind review

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 method 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.

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 (Pathak et al. (2014); Iizuka et al. (2017)), super resolution (Dong et al. (2016); Kim et al. (2016)), colorization (Zhang et al. (2016; 2017)) and style transfer (Gatys et al. (2016); Huang & Belongie (2017)). When paired samples are given, the mapping model can be trained in a supervised manner using a conditional generative model (Isola et al. (2017); Li et al. (2017a); Wang et al. (2018)) or a simple regression model (Larsson et al. (2016); Long et al. (2015); Zhang et al. (2016)). In unsupervised settings where no paired data is available, multiple works (Anoosheh et al. (2018); Choi et al. (2018); Huang et al. (2018); Kim et al. (2017); Liu et al. (2017); Royer et al. (2017); Taigman et al. (2017); Yi et al. (2017); Zhu et al. (2017)) successfully have translated images using shared latent space (Liu et al. (2017)) and cycle consistency assumptions (Kim et al. (2017); Zhu et al. (2017)). These works have been further developed to handle the multi-modality of the task (Huang et al. (2018)).

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 (Huang et al. (2018); Liu et al. (2017)). In addition, existing methods such as DRIT (Lee et al. (2018)) 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

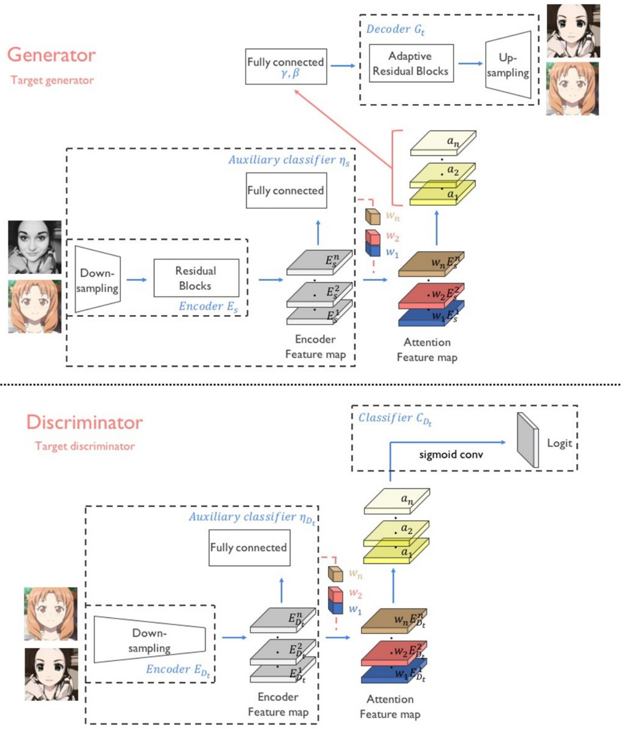

Figure 1: The model architecture of U-GAT-IT. The detailed notations are described in Section Model

manner. 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 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) (Nam & Kim (2018)), 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 Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization

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.

2.1 Model

2.1.1 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 (Zhou et al. (2016)), 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 (Huang & Belongie (2017); Nam & Kim (2018)), we equip the residual blocks with AdaLIN whose parameters are dynamically computed by a fully connected layer from the attention map.

$$
\begin{aligned}
AdaLIN(a,\gamma,\beta) &= \gamma \cdot (\rho \cdot \hat{a_I} + (1 - \rho) \cdot \hat{a_L}) + \beta, \\
\hat{a_I} &= \frac{a - \mu_I}{\sqrt{\sigma_I^2 + \epsilon}}, \hat{a_L} = \frac{a - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}}, \\
\rho &\leftarrow clip_{[0,1]}(\rho - \tau \Delta \rho)
\end{aligned}
\tag{1}
$$

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) (Li et al. (2017b)), but the computational cost is high due to the calculation of the covariance matrix and matrix inverse. Although, the AdaIN (Huang & Belongie (2017)) 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 (Ba et al. (2016)) 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.

2.1.2 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 $C_{D_t}(a_{D_t}(x))$.

2.2 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 (Mao et al. (2017)) objective for stable training.

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

$$L_{lsgan}^{s\to t} = (\mathbb{E}_{x\sim X_t}[(D_t(x))^2] + \mathbb{E}_{x\sim X_s}[(1 - D_t(G_{s\to t}(x)))^2]). \tag{2}$$

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:

$$L_{cycle}^{s\to t} = \mathbb{E}_{x\sim X_s}[|x - G_{t\to s}(G_{s\to t}(x)))|_1]. \tag{3}$$

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.

$$L_{identity}^{s\to t} = \mathbb{E}_{x\sim X_t}[|x - G_{s\to t}(x)|_1]. \tag{4}$$

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:

$$L_{cam}^{s\to t} = -(\mathbb{E}_{x\sim X_s}[log(\eta_s(x))] + \mathbb{E}_{x\sim X_t}[log(1 - \eta_s(x))]), \tag{5}$$

$$L_{cam}^{D_t} = \mathbb{E}_{x\sim X_t}[(\eta_{D_t}(x))^2] + \mathbb{E}_{x\sim X_s}[(1 - \eta_{D_t}(G_{s\to t}(x))^2]. \tag{6}$$

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

$$\min_{G_{s\to t}, G_{t\to s}, \eta_s, \eta_t} \max_{D_s, D_t, \eta_{D_s}, \eta_{D_t}} \lambda_1 L_{lsgan} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam}, \tag{7}$$

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

3 Experiments

3.1 Baseline model

We have compared our method with various models including CycleGAN (Zhu et al. (2017)), UNIT (Liu et al. (2017)), MUNIT (Huang et al. (2018)), DRIT (Lee et al. (2018)), AGGAN (Mejjati et al. (2018)), and CartoonGAN (Chen et al. (2018)). All the baseline methods are implemented using the author's code.

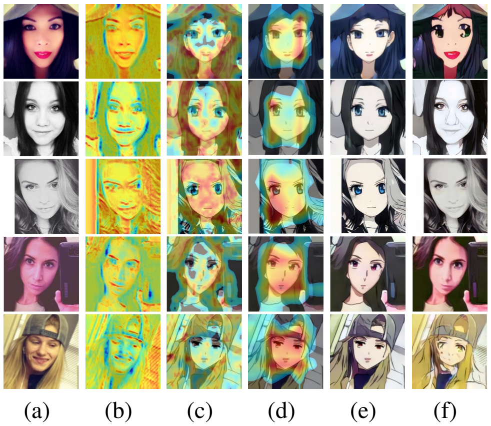

Figure 2: Visualization of the attention maps and their effects shown in the ablation experiments: (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminator, respectively. (e) Our results with CAM, (f) Results without CAM.

3.2 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. See Appendix B for each dataset for our experiments.

3.3 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 comparing our model with other models are included in the supplementary materials.

3.3.1 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).

3.3.2 AdaLIN analysis

As described in Appendix A, 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.

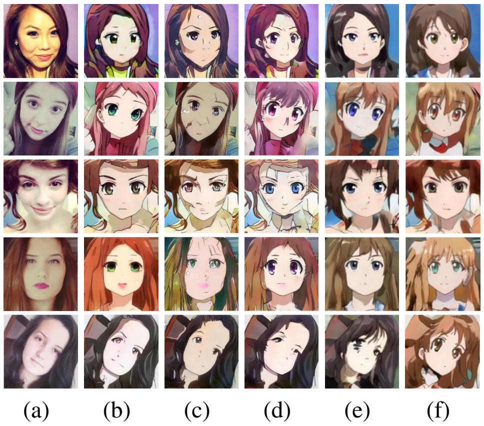

Figure 3: Comparison of the results using each normalization function: (a) Source images, (b) Our results, (c) Results only using IN in decoder with CAM, (d) Results only using LN in decoder with CAM, (e) Results only using AdaIN in decoder with CAM, (f) Results only using GN in decoder with CAM.

Table 1: Kernel Inception Distance×100±std.×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	selfie2anime	anime2selfie
U-GAT-IT	**11.61** ± **0.57**	**11.52** ± **0.57**
U-GAT-IT w/ IN	13.64 ± 0.76	13.58 ± 0.8
U-GAT-IT w/ LN	12.39 ± 0.61	13.17 ± 0.8
U-GAT-IT w/ AdaIN	12.29 ± 0.78	11.81 ± 0.77
U-GAT-IT w/ GN	12.76 ± 0.64	12.30 ± 0.77
U-GAT-IT w/o CAM	12.85 ± 0.82	14.06 ± 0.75
U-GAT-IT w/o G_CAM	12.33 ± 0.68	13.86 ± 0.75
U-GAT-IT w/o D_CAM	12.49 ± 0.74	13.33 ± 0.89

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 3 (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 3 (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 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 3 (e), (f) are the results of using the AdaIN and Group Normalization (GN) (Wu & He (2018)) respectively, and our methods are showing better results compared to these.

Also, as shown 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) (Bińkowski et al. (2018)). Our model achieves the lowest KID values. Even if the attention mod-

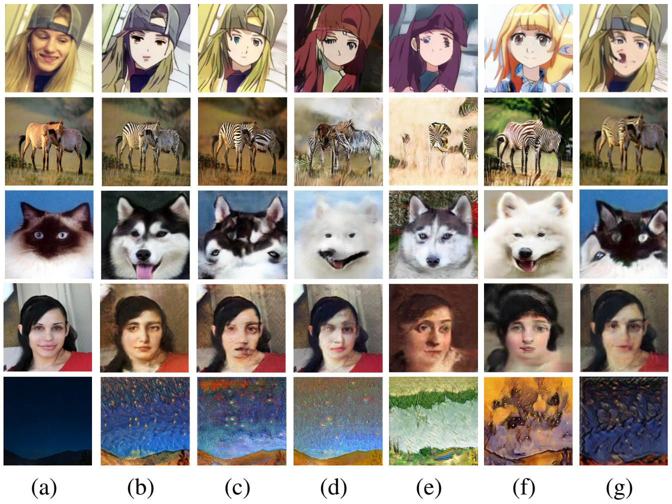

Figure 4: Visual comparisons on the five datasets. From top to bottom: selfie2anime, horse2zebra, cat2dog, photo2portrait, and photo2vangogh. (a)Source images, (b)U-GAT-IT, (c)CycleGAN, (d)UNIT, (e)MUNIT, (f)DRIT, (g)AGGAN

ule 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.

3.3.3 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 4, 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. 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 more flexible.

3.3.4 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 (Szegedy et al. (2016)). In contrast to the Fréchet Inception Distance (Heusel et al. (2017)), 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 (Mejjati et al. (2018)). 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

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

Model	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$\times$100$\pm$std.$\times$100 for difference image translation mode. Lower is better.

Model	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	**1.20 ± 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
AGGAN	14.63 ± 0.55	7.58 ± 0.71	9.84 ± 0.79	2.33 ± 0.36	6.95 ± 0.33
CartoonGAN	15.85 ± 0.69	-	-	-	-
Model	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	3.30 ± 0.47	9.53 ± 0.35
DRIT	14.85 ± 0.60	10.98 ± 0.55	10.86 ± 0.24	4.76 ± 0.72	7.72 ± 0.34
AGGAN	12.72 ± 1.03	8.80 ± 0.66	9.45 ± 0.64	2.19 ± 0.40	5.85 ± 0.31

no big difference from the lowest score. Also, unlike UNIT and MUNIT, we can see that the source → target, target → source translations are both stable. U-GAT-IT shows even lower KID than the recent attention-based method, AGGAN. AGGAN yields poor performance for the transformation with shape change such as dog2cat and anime2selfie unlike the U-GAT-IT, the attention module of which focuses on distinguishing not between background and foreground but differences between two domains. CartoonGAN, as shown in the supplementary materials, has only changed the overall color of the image to an animated style, but compared to selfie, the eye, which is the biggest characteristic of animation, has not changed at all. Therefore, CartoonGAN has the higher KID.

4 Conclusions

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

Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition Workshops, pp. 783–790, 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9465–9474, 2018.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2): 295–307, 2016.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision*, pp. 172–189, 2018.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):107, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646–1654, 2016.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, pp. 1857–1865, 2017. URL http://proceedings.mlr.press/v70/kim17a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6980.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision*, pp. 577–593. Springer, 2016.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision*, pp. 35–51, 2018.

Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems*, pp. 5501–5509, 2017a.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pp. 386–396, 2017b.

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

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2813–2821. IEEE, 2017.

Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Advances in Neural Information Processing Systems*, pp. 3697–3707, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems*, pp. 2563–2572, 2018. URL http://papers.nips.cc/paper/7522-batch-instance-normalization-for-adaptively-style-invariant-neural-networks.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2014.

Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Moressi, Forrester Cole, and Kevin Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. *arXiv preprint arXiv:1711.05139*, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sk2Im59ex.

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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, September 2018.

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2849–2857, 2017.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*, pp. 649–666. Springer, 2016.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, Tianhe Yu, and Alexei A. Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 36(4):119:1–119:11, 2017. doi: 10.1145/3072959.3073703. URL `https://doi.org/10.1145/3072959.3073703`.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929. IEEE, 2016.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

A Implementation Details

A.1 Network architecture

The network architectures of U-GAT-IT are shown in Table 4, 5, and 6. 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 (Wu & He (2018)). 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 (Miyato et al. (2018)) is used for the discriminator. We employ two different scales of PatchGAN (Isola et al. (2017)) 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.

A.2 Training

All models are trained using Adam (Kingma & Ba (2015)) 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.

B Dataset Details

selfie2anime The selfie dataset contains 46,836 selfie images annotated with 36 different attributes. 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].

horse2zebra and photo2vangogh These datasets are used in CycleGAN (Zhu et al. (2017)). 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 (Lee et al. (2018)). 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.

C Additional experimental results

In addition to the results presented in the paper, we show supplement generation results for the five datasets in Figs 5, 6, 7, 8, 9, 10, 11, and 12.

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

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

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

Table 4: The detail of generator architecture.

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

Table 6: The detail of global discriminator.

Part	Input → Output Shape	Layer Information
6*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
3*CAM of Discriminator	2*($\frac{h}{32}$, $\frac{w}{32}$, 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

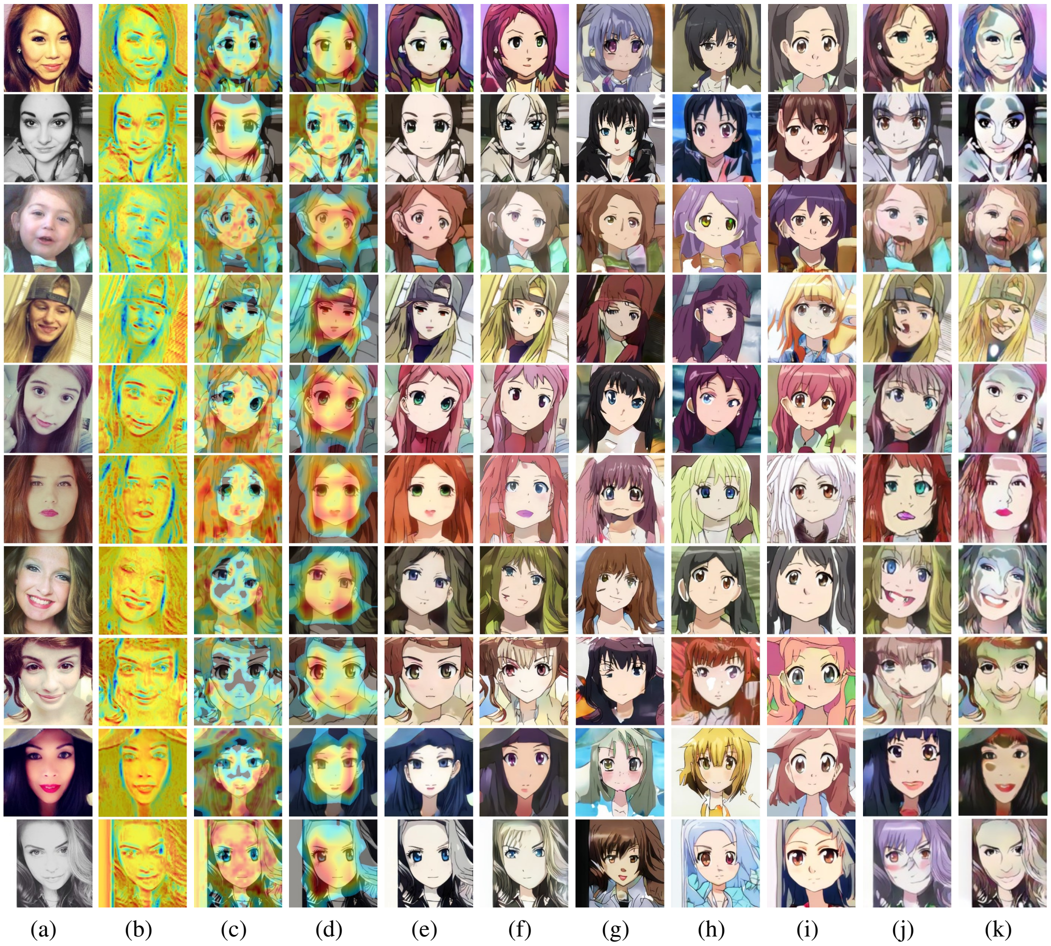

Figure 5: Visual comparisons of the selfie2anime with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN Zhu et al. (2017), (g) UNIT Liu et al. (2017), (h) MUNIT Huang et al. (2018), (i) DRIT Lee et al. (2018), (j) AGGAN Mejjati et al. (2018), (k) CartoonGAN Chen et al. (2018).

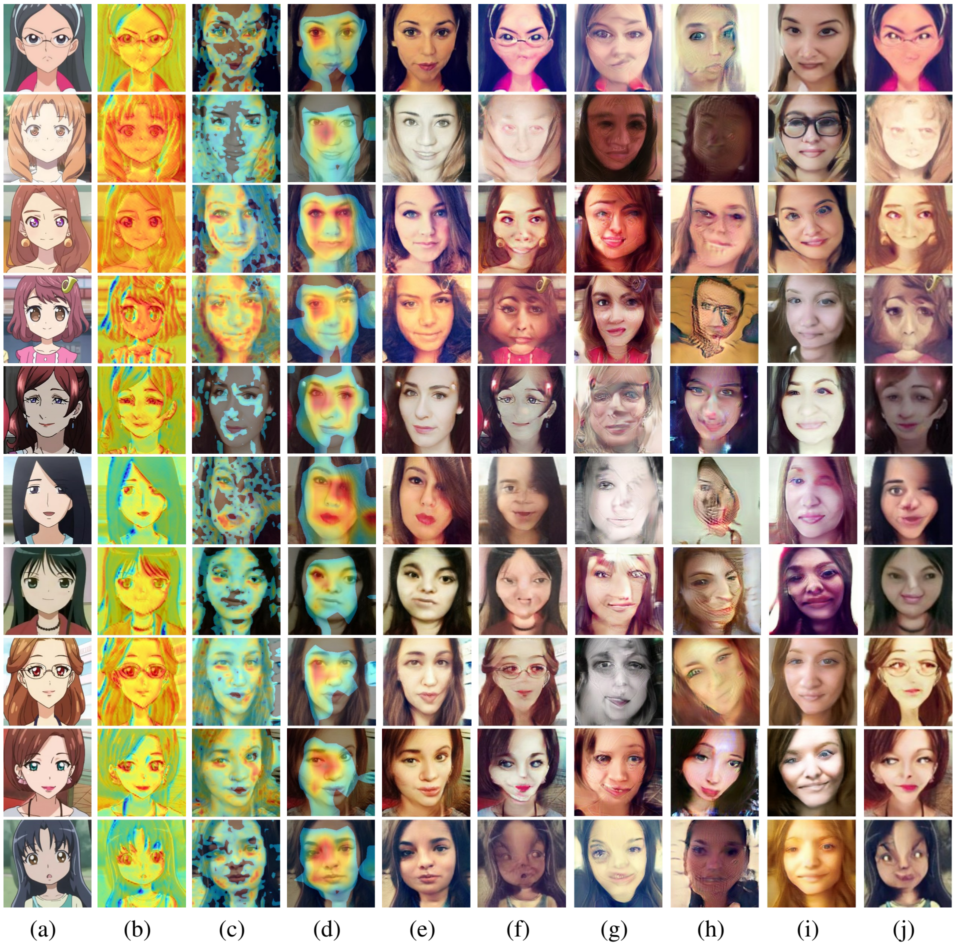

Figure 6: Visual comparisons of the anime2selfie with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.

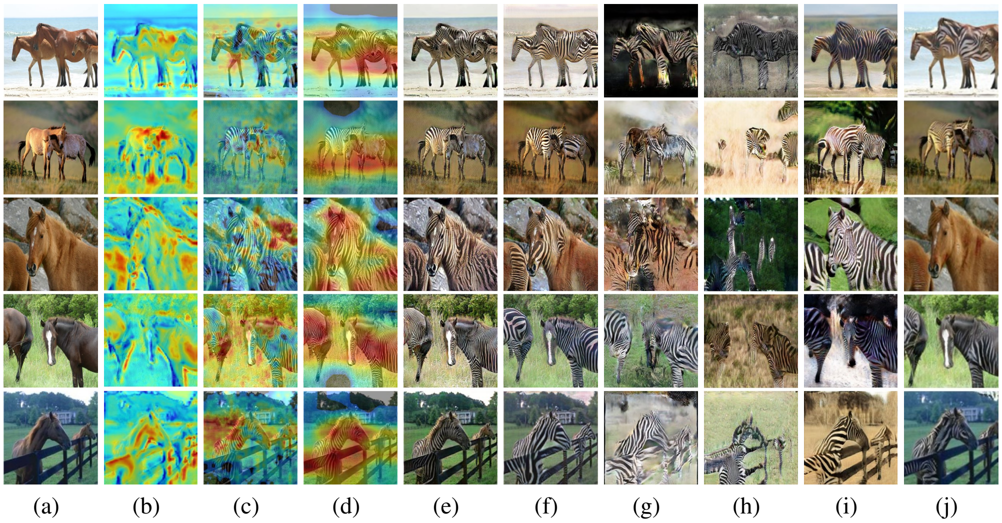

Figure 7: Visual comparisons of the horse2zebra with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT, (j) AGGAN

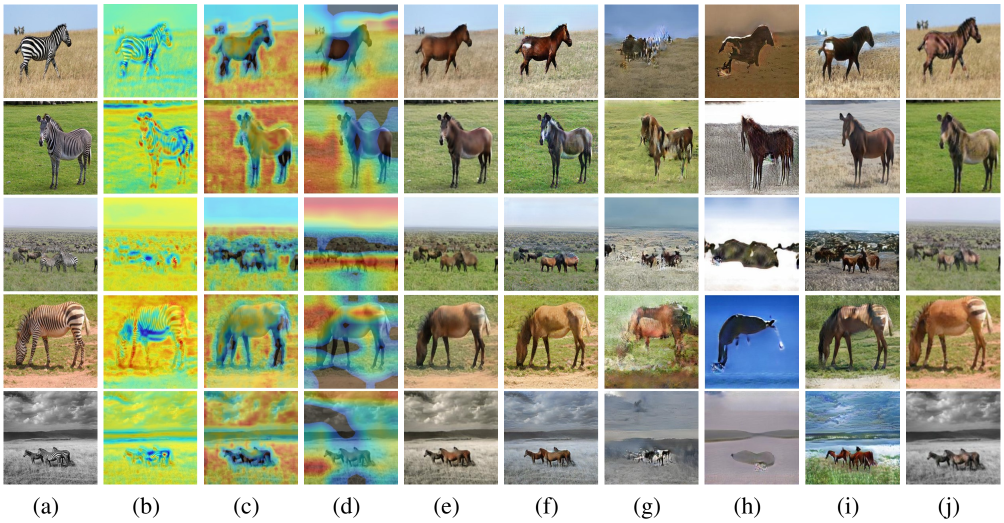

Figure 8: Visual comparisons of the zebra2horse with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.

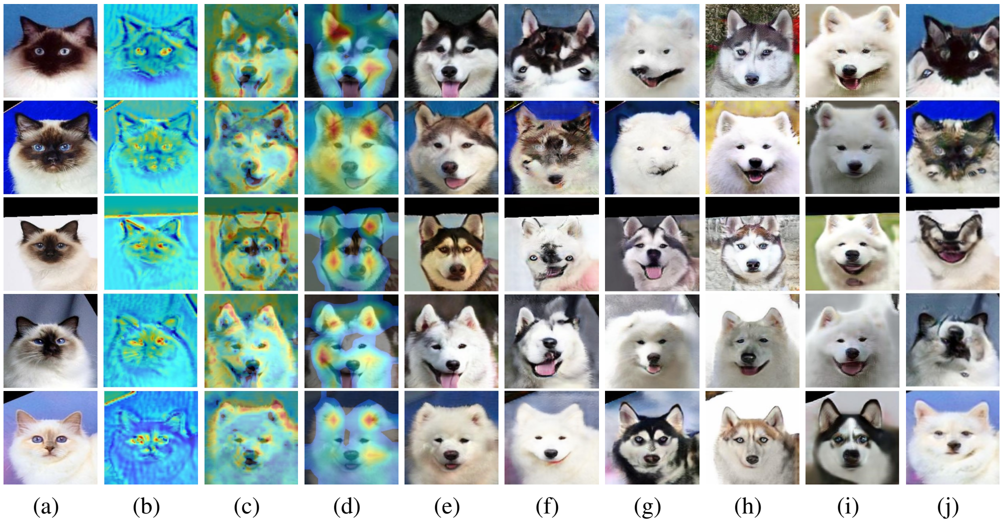

Figure 9: Visual comparisons of the cat2dog with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.

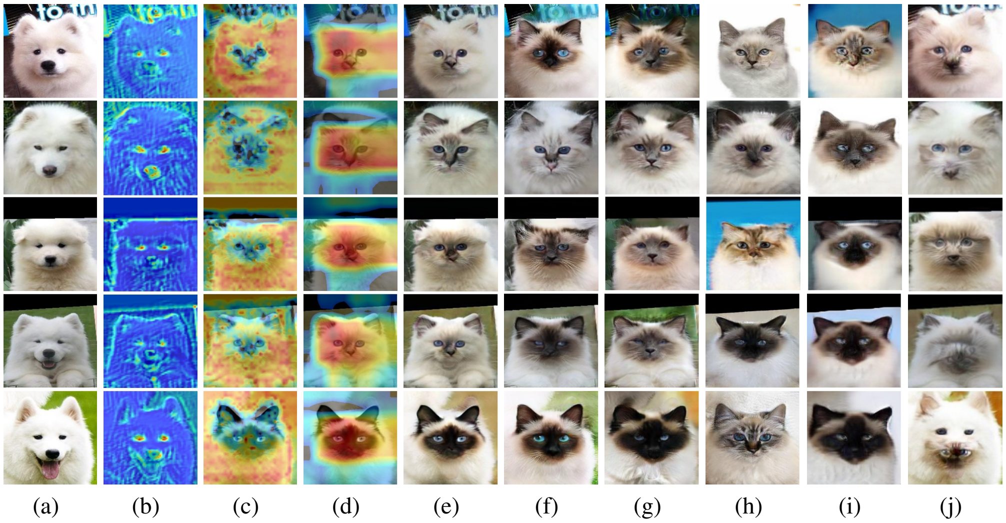

Figure 10: Visual comparisons of the dog2cat with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.

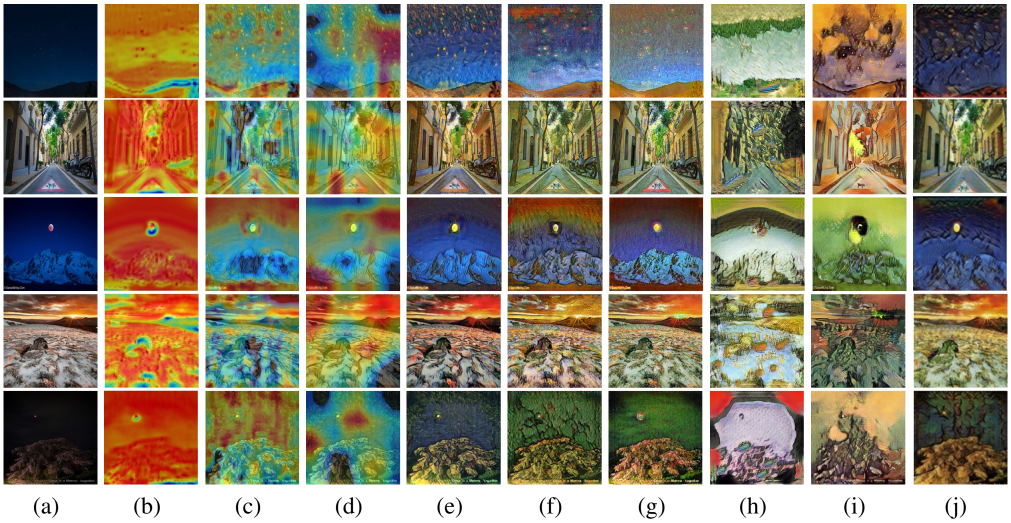

Figure 11: Visual comparisons of the photo2vangogh with attention features maps. (a) Source images, (b) Attention map of the generation, (c-d) Local and global attention maps of the discriminators, respectively, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.

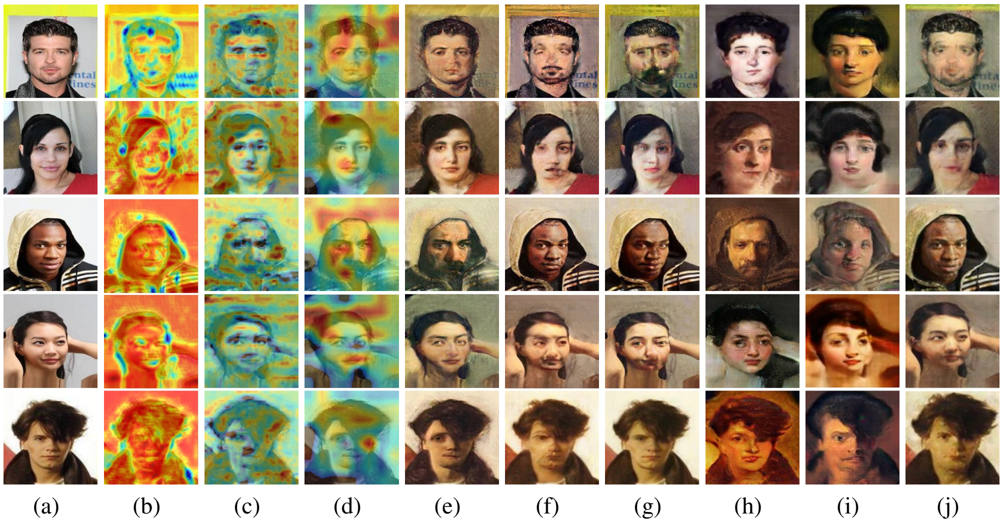

Figure 12: Visual comparisons of the photo2portrait with attention features maps. (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminators, respectively, (e) Our results, (f) CycleGAN, (g) UNIT, (h) MUNIT, (i) DRIT (j) AGGAN.