SagaGAN: Style Applied using Gram matrix Attribution based on StarGAN v2

Yongseon Yoo[1]
rs21140@hanyang.ac.kr

Seonggyu Kim[2]
fehur@hanyang.ac.kr

Jong-Min Lee[1, 2, 3]
ljm@hanyang.ac.kr

[1] Department of Artificial Intelligence
Hanyang University
Seoul, Korea

[2] Department of Electronic Engineering
Hanyang University
Seoul, Korea

[3] Department of Biomedical Engineering
Hanyang University
Seoul, Korea

Abstract

Image-to-image translation aims to convert an image from one domain to another while preserving its content. AdaIN (Adaptive Instance Normalization) is a widely used style application method, but it may not fully capture the fine-grained visual characteristics of complex styles. We propose SagaGAN, a novel approach that combines the gram matrix with AdaIN to better capture and transfer style information. We introduce two loss functions: G1 loss and G2 loss, which focus on the differences between gram matrices of the style, generated, and input images. These losses enable SagaGAN to learn richer style information. Additionally, we incorporate a perceptual loss alongside the cycle consistency loss to maintain a balance between style application and content preservation. Experimental results demonstrate that SagaGAN effectively applies style information, leading to improved image generation performance compared to existing models. By leveraging the gram matrix to capture complex style characteristics while preserving content, SagaGAN enhances the style transfer capabilities of models like StarGAN v2.

1 Introduction

Image-to-image translation, which aims to convert an image from one domain to another while preserving its original content, has been a key research area in computer vision [7]. It has various applications such as style transfer, virtual try-on, and data augmentation [15]. However, effectively applying the desired style to an image while maintaining its content remains a challenging task.

Previous methods for image-to-image translation, such as CycleGAN [21] and UNIT [14], require multiple generators and additional networks for multi-domain translation. StarGAN v2 [1] addressed this limitation by training a single generator in a multi-domain setting and demonstrated excellent performance in various image synthesis tasks within a single domain, such as generating female images with different hair colors and skin tones. The style application method used in StarGAN v2 is AdaIN (Adaptive Instance Normalization) [6].

Prior to the introduction of AdaIN (Adaptive Instance Normalization) [6], style transfer methods such as the pioneering work by Gatys et al. [3] and its variants [4, 8, 13, 19] focused on iteratively optimizing the generated image to match the style statistics of the target style image. These methods often relied on the gram matrix, which captures the correlations between different features, to represent the style information. While effective, these approaches were computationally expensive and required iterative optimization for each input image.

AdaIN, on the other hand, provides a more efficient way to apply style by adjusting the mean and variance of the content features to match those of the style features. AdaIN has been widely adopted in various style transfer and image synthesis tasks [11, 12, 17, 22] due to its effectiveness and efficiency.

However, AdaIN may not be sufficient to capture the fine-grained visual characteristics of complex styles. AdaIN adjusts the scale and bias values corresponding to the mean and variance of each convolutional layer's output, but these simple statistical properties may not fully represent the intricate elements that constitute an image's style such as textures and patterns.

In this paper, we propose a novel approach to the image-to-image translation task. Recognizing the importance of style, we introduce a method that combines the gram matrix approach with AdaIN to better capture complex style characteristics. By incorporating the gram matrix, which considers the correlations between features, alongside AdaIN's efficient style application, our method can more effectively represent and apply the intricate elements that constitute an image's style, such as textures and patterns.

We integrate two losses in the training process: G1 loss, which minimizes the difference between the gram matrices of the style image and the image generated by the generator, and G2 loss, which reduces the difference between the gram matrices obtained by feeding the input image and the style image into the encoder part of the generator. By introducing G1 and G2 losses, we increase the proportion of losses that account for style in the overall loss function, enabling the model to learn richer style information. Furthermore, to better preserve content alongside the cycle consistency loss, we incorporate a perceptual loss that reduces the difference between the input and output images.

Our experimental results demonstrate that the proposed method effectively applies style information, leading to improved image generation performance compared to existing models. This enhancement is quantitatively measured using evaluation metrics such as FID (Fréchet Inception Distance) [5] and LPIPS (Learned Perceptual Image Patch Similarity) [20], which assess the quality and perceptual similarity of generated images.

The purpose of this paper is to present SagaGAN, a novel approach to the image-to-image translation task. By combining the gram matrix approach with AdaIN and introducing G1 and G2 loss functions in the StarGAN v2 model, our method can better capture complex style characteristics. Additionally, the incorporation of a perceptual loss helps to maintain a balance between style application and content preservation, ultimately enhancing image generation performance.

2 Related Work

2.1 StarGAN v2

The main components of StarGAN v2 are the generator, mapping network, style encoder, and discriminator [1]. The generator takes a source image and a style code as input, and synthesizes an output image using AdaIN (Adaptive Instance Normalization) to inject the style information. The mapping network transforms random Gaussian noise into a style code, while the style encoder extracts the style code from a given reference image.

StarGAN v2 employs several loss functions to train the model, including the adversarial loss, style reconstruction loss, style diversification loss, and cycle consistency loss [1]. These losses work together to ensure that the generated images are realistic, diverse, and consistent with the source images and target domains.

2.2 AdaIN

AdaIN (Adaptive Instance Normalization) is a technique introduced by Huang and Belongie for arbitrary style transfer [6]. It normalizes the feature maps of the content image to match the channel-wise mean and variance of the style image, effectively transferring the style information while preserving the content structure. The AdaIN operation is defined as [6]:

$$\mathrm{AdaIN}(x,y) = \sigma(y) \cdot \frac{(x-\mu(x))}{\sigma(x)} + \mu(y) \tag{1}$$

where x is the content feature map, y is the style feature map, μ and σ represent the channel-wise mean and standard deviation, respectively. This operation first normalizes the content feature map x to have zero mean and unit variance, then scales it with the standard deviation of the style feature map $\sigma(y)$ and shifts it with the mean of the style feature map $\mu(y)$. This alignment of feature statistics enables the transfer of style information from the style image to the content image.

2.3 Gram Matrix

The Gram matrix, originally introduced in the context of style transfer by Gatys et al. [3, 4], is a key concept in capturing the style of an image. It is a measure of the correlations between different features in a given layer of a Convolutional Neural Network (CNN).

Mathematically, the Gram matrix $G \in R^{C\times C}$ of a given feature map $F \in R^{C\times HW}$, where C is the number of channels, H is the height, and W is the width, is defined as [3, 4]:

$$G_{ij} = \sum_k F_{ik} \cdot F_{jk} \tag{2}$$

where F_{ik} is the activation of the i-th channel at position k in the vectorized feature map, and F_{jk} is the activation of the j-th channel at the same position.

Intuitively, the Gram matrix captures the correlations between different channels in a given layer. The diagonal entries of the Gram matrix represent the self-correlations, i.e., the average of the squared activations of each feature. The off-diagonal entries represent the cross-correlations between different features. By capturing these correlations, the Gram matrix provides a summary statistic of the feature activations in a given layer.

3 Method

3.1 Proposed Objective

The main objective of SagaGAN is to enhance the style transfer capabilities of StarGAN v2 [1] by incorporating the gram matrix, which captures the correlations between different

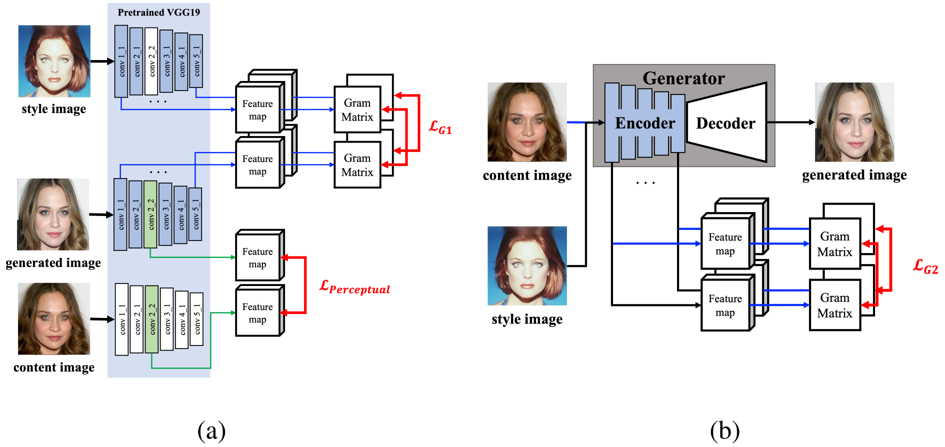

Figure 1: Computation of Proposed G1 Loss, Perceptual Loss (a), and G2 Loss (b) in SagaGAN. Exclude `conv2_2` feature map when calculating G1 Loss (a). Details are described in Sections 3.1.1, 3.1.2 and 3.1.3.

feature maps and is considered to account for style information[3, 4, 9, 10]. To achieve this objective, SagaGAN introduces two novel losses: G1 loss and G2 loss. By integrating the gram matrix into the style application process, SagaGAN aims to better capture and transfer the fine-grained visual characteristics of complex styles, which may not be fully represented by the simple statistical properties used in AdaIN.

3.1.1 G1 Loss

The G1 loss plays a crucial role in SagaGAN by guiding the generator to learn and apply the style of the reference image to the generated output. As illustrated in Figure 1(a), the G1 loss is calculated by minimizing the discrepancy between the Gram matrices of the style image and the generated image, which are obtained from the VGG-19 network [18].

To compute the G1 loss, both the style image and the generated image are first passed through a pre-trained VGG-19 network, which serves as a feature extractor. The features are extracted from the first convolutional layer at five different levels of the network, specifically `conv1_1`, `conv2_1`, `conv3_1`, `conv4_1`, and `conv5_1`. These features capture style information at various scales and levels of abstraction.

The Gram matrix is then calculated for each of these five feature maps. The Gram matrix encodes the correlations between different features within a layer, effectively capturing the style information[3, 4, 9, 10].

$$L_{G1} = \frac{1}{2}\sum_{l=1}^{L}\left(Gram^{l}_{G(x,s)} - Gram^{l}_{sty}\right)^{2} \tag{3}$$

Where, L is the total number of layers ($L = 5$ in this case), $Gram_G(x,s)^l$ represents the Gram matrix of the generated image at layer l, and $Gram^l_{sty}$ denotes the Gram matrix of the style image at layer l.

By minimizing the G1 loss, the style encoder, generator learns to adjust the style of the generated image to match that of the reference style image. This is accomplished by aligning the feature correlations, as captured by the Gram matrices, across multiple scales. The multi-scale nature of this loss enables the generator to capture both fine-grained and high-level style information, resulting in more effective style transfer.

3.1.2 G2 Loss

Unlike the G1 loss, which is used to train the Generator, Mapping Network, and Style Encoder, the primary purpose of the G2 loss is to encourage the generator's encoder to better capture and reflect style information.

The motivation for introducing the G2 loss is to address the limitations of the encoder in the original generator architecture. In the base model, the encoder primarily encodes content information, and style application is mainly handled by the decoder. However, by integrating the G2 loss into the training process, the encoder is encouraged to learn and apply style information, allowing both the encoder and decoder to contribute to the style transfer process, enhancing the generator's overall style transfer capabilities.

As shown in Figure 1(b), the G2 loss is calculated by minimizing the difference between the Gram matrices obtained from the feature maps of the content image and the style image passing through the encoder part of the generator.

$$L_{G2} = \frac{1}{2}\sum_{l=1}^{L}\left(Gram_x^l - Gram_{sty}^l\right)^2 \tag{4}$$

Where, L is the total number of layers, $Gram_x^l$ represents the Gram matrix of the content image at layer l, and $Gram_{sty}^l$ denotes the Gram matrix of the style image at layer l.

By minimizing the G2 loss, the encoder learns to style. This, combined with the decoder's style application abilities, results in improved style application in the generated images.

3.1.3 Perceptual Loss

The addition of G1 and G2 losses to the original StarGAN v2 framework increases the proportion of style-related losses in the overall objective function. While style transfer is a crucial aspect of image-to-image translation, maintaining the content of the input image is equally important. In StarGAN v2, the cycle consistency loss is the main component responsible for preserving content. However, relying solely on the cycle consistency loss may not be sufficient to achieve a satisfactory balance between style transfer and content preservation.

Instead of simply adjusting the weight of the cycle consistency loss, we propose incorporating a separate perceptual loss that encourages content preservation in a different manner, as illustrated in Figure 1(a). The perceptual loss minimizes the difference between the feature maps of the input image and the generated image, ensuring that the generator maintains the content information during the stylization process. It is defined as:

$$\mathcal{L}_{\text{perceptual}} = \left(F_{G(x,s)} - F_x\right)^2 \tag{5}$$

Where F represents the feature maps extracted from a specific layer of a pre-trained CNN. In our implementation, we use the VGG-19 network [18], which has been widely used for perceptual losses in various image generation tasks. We extract the feature maps from the `conv2_2` layer of VGG-19 to compute the perceptual loss, as suggested by the findings in [3]. This work demonstrated that using feature maps from earlier layers in the network leads to stronger content preservation.

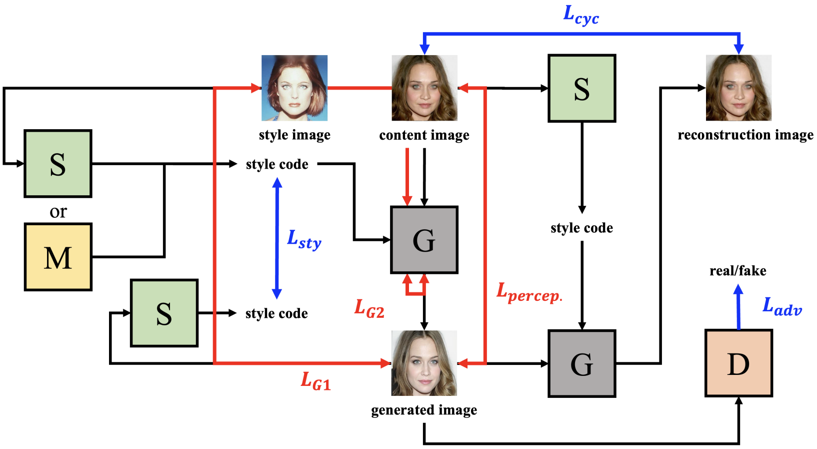

Figure 2: The Architecture of SagaGAN. Detailed explanations of each component are provided in Sections 3.2.

3.2 Network Architecture and Overall Objective Function

As illustrated in Figure 3, SagaGAN adopts the basic network architecture of StarGAN v2 [1]. The fundamental structure consists of a generator G, a mapping network M, a style encoder S, and a discriminator D. The generator G takes a content image and a style code as input and synthesizes a style-transferred image. The style code can be obtained either from the mapping network M, which transforms a random latent code z into a style code, or from the style encoder S, which extracts a style code from a given reference image. The discriminator D is trained to distinguish between real and fake images and predict the corresponding domain.

The objective function of SagaGAN is designed to facilitate effective style transfer while preserving the content of the input image by applying the proposed G1, G2, and perceptual losses (red arrows in Figure 3) to the basic objective function of StarGAN v2 (blue arrows in Figure 3).

The overall objective function of SagaGAN can be expressed as:

$$\min_{G,F,E}\max_{D}\left(L_{adv}+\lambda_{sty}L_{sty}-\lambda_{ds}L_{ds}+\lambda_{cyc}L_{cyc}+\lambda_{G1}L_{G1}+\lambda_{perc.}L_{perc.}+\lambda_{G2}L_{G2}\right) \tag{6}$$

Where L_{adv} is the adversarial loss, L_{sty} is the style reconstruction loss, L_{ds} is the style diversification loss, L_{cyc} is the cycle consistency loss, L_{G1} is the G1 loss, $L_{perc.}$ is the perceptual loss, and L_{G2} is the G2 loss. The λ terms are hyperparameters that control the relative importance of each loss component.

The style reconstruction loss L_{sty} encourages the generator to produce images that match the style of the reference image. It is computed as the $L1$ distance between the style code

$E_{\tilde{y}}(G(x,\tilde{s}))$ extracted from the generated image $G(x,\tilde{s})$ and the style code of the reference image $\tilde{s}$:

$$L_{sty} = E_{(x,\tilde{y},z)}\left[\|\tilde{s} - E_{\tilde{y}}(G(x,\tilde{s}))\|_1\right] \tag{7}$$

The style diversification loss L_{ds} promotes diversity in the generated images by maximizing the distance between the style codes $\tilde{s}_1$ and $\tilde{s}_2$ of different generated images:

$$L_{ds} = E_{(x,\tilde{y},z_1,z_2)}\left[\|G(x,\tilde{s}_1) - G(x,\tilde{s}_2)\|_1\right] \tag{8}$$

The cycle consistency loss L_{cyc} ensures that the generated image preserves the content of the input image. It is calculated as the $L1$ distance between the input image x and the reconstructed image $G(G(x,\tilde{s}),\hat{s})$:

$$L_{cyc} = E_{(x,y,\tilde{y},z)}\left[\|x - G(G(x,\tilde{s}),\hat{s})\|_1\right] \tag{9}$$

The G1 loss (Section 3.1.1) and G2 loss (Section 3.1.2) assist in the style application of the generated images during the style transfer process, while the perceptual loss (Section 3.1.3) helps preserve the content of the input image during the style transfer process. By optimizing this objective function, SagaGAN learns to generate high-quality stylized images that effectively capture the style of the reference image while preserving the content of the input image.

4 Experiments

4.1 Experimental Setup

We utilize the CelebA-HQ and AFHQ[2] datasets provided by StarGAN v2, along with the additional FFHQ[16] dataset for our experiments. All datasets are resized to a resolution of 256 × 256 pixels.

We assess the performance of SagaGAN using two widely adopted metrics: Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS). FID measures the quality of the generated images by comparing the distribution of features extracted from the generated images and real images using the Inception-v3 network. A lower FID score indicates better image quality and closer similarity to real images. LPIPS, on the other hand, evaluates the diversity of the generated images by measuring the perceptual similarity between pairs of images. A higher LPIPS score suggests greater diversity among the generated images.

4.2 Quantitative Results

Table 1 presents the quantitative results of SagaGAN compared to StarGAN v2 on various datasets, evaluated using Fréchet Inception Distance (FID) for image quality and Learned Perceptual Image Patch Similarity (LPIPS) for diversity. SagaGAN consistently outperforms StarGAN v2 in terms of FID, particularly in the female→male task on CelebA-HQ and the cat→wild and dog→wild tasks on AFHQ, where SagaGAN achieves significantly lower FID scores. These improvements can be attributed to SagaGAN's combination of the gram matrix approach with AdaIN, enabling better style transfer. SagaGAN also generates more diverse images, as indicated by higher LPIPS scores on AFHQ and FFHQ. The results

Dataset	Method	Task	latent		reference	
			FID ↓	LPIPS ↑	FID ↓	LPIPS ↑
CelebA-HQ	StarGAN v2	male → female	10.55	**0.444**	**19.27**	0.374
		female → male	18.62	0.460	26.70	0.401
		Mean	14.59	**0.452**	22.98	0.388
	SagaGAN	male → female	**9.73**	0.436	19.49	**0.380**
		female → male	**17.32**	**0.462**	**26.15**	**0.401**
		Mean	**13.53**	0.449	**22.82**	**0.391**
AFHQ	StarGAN v2	dog → cat	**6.67**	0.411	7.01	0.413
		cat → dog	37.97	0.413	41.53	0.440
		wild → cat	**8.31**	**0.456**	**7.73**	**0.416**
		cat → wild	33.97	0.452	39.10	0.421
		wild → dog	**31.91**	0.445	**36.89**	0.432
		dog → wild	33.84	0.434	40.56	0.408
		Mean	25.44	0.435	28.80	0.422
	SagaGAN	dog → cat	7.31	**0.420**	**6.99**	**0.418**
		cat → dog	**35.60**	**0.458**	**38.60**	**0.443**
		wild → cat	8.59	0.416	7.81	0.415
		cat → wild	**16.65**	**0.462**	**18.04**	**0.444**
		wild → dog	33.50	**0.452**	37.62	**0.433**
		dog → wild	**16.14**	**0.464**	**19.70**	**0.444**
		Mean	**19.63**	**0.445**	**21.46**	**0.433**
FFHQ	StarGAN v2	male → female	22.36	0.067	21.78	0.092
		female → male	26.08	0.061	25.83	0.081
		Mean	24.22	0.064	23.80	0.086
	SagaGAN	male → female	**18.78**	**0.133**	**18.42**	**0.137**
		female → male	**23.17**	**0.136**	**23.50**	**0.134**
		Mean	**20.98**	**0.135**	**20.96**	**0.136**

Table 1: Quantitative comparison of StarGAN v2 and SagaGAN on various datasets using FID and LPIPS metrics. The best results for each dataset are shown in bold.

demonstrate SagaGAN's effectiveness in generating high-quality and diverse images across various datasets and tasks.

4.3 Qualitative Results

Figure 3 presents a qualitative comparison of image-to-image translation results between StarGAN v2 and SagaGAN on the CelebA-HQ, AFHQ, and FFHQ datasets. The generated images demonstrate that SagaGAN achieves noticeably improved style transfer compared to the baseline StarGAN v2 model. On the CelebA-HQ and FFHQ datasets, SagaGAN exhibits a more comprehensive enhancement in applying styles to human skin tones and hair representations compared to the existing model. Additionally, as observed in the AFHQ dataset, where images of dogs and cats are translated to the "wild" domain, SagaGAN effectively applies the desired style across domains, surpassing the performance of the baseline model in cross-domain style transfer.

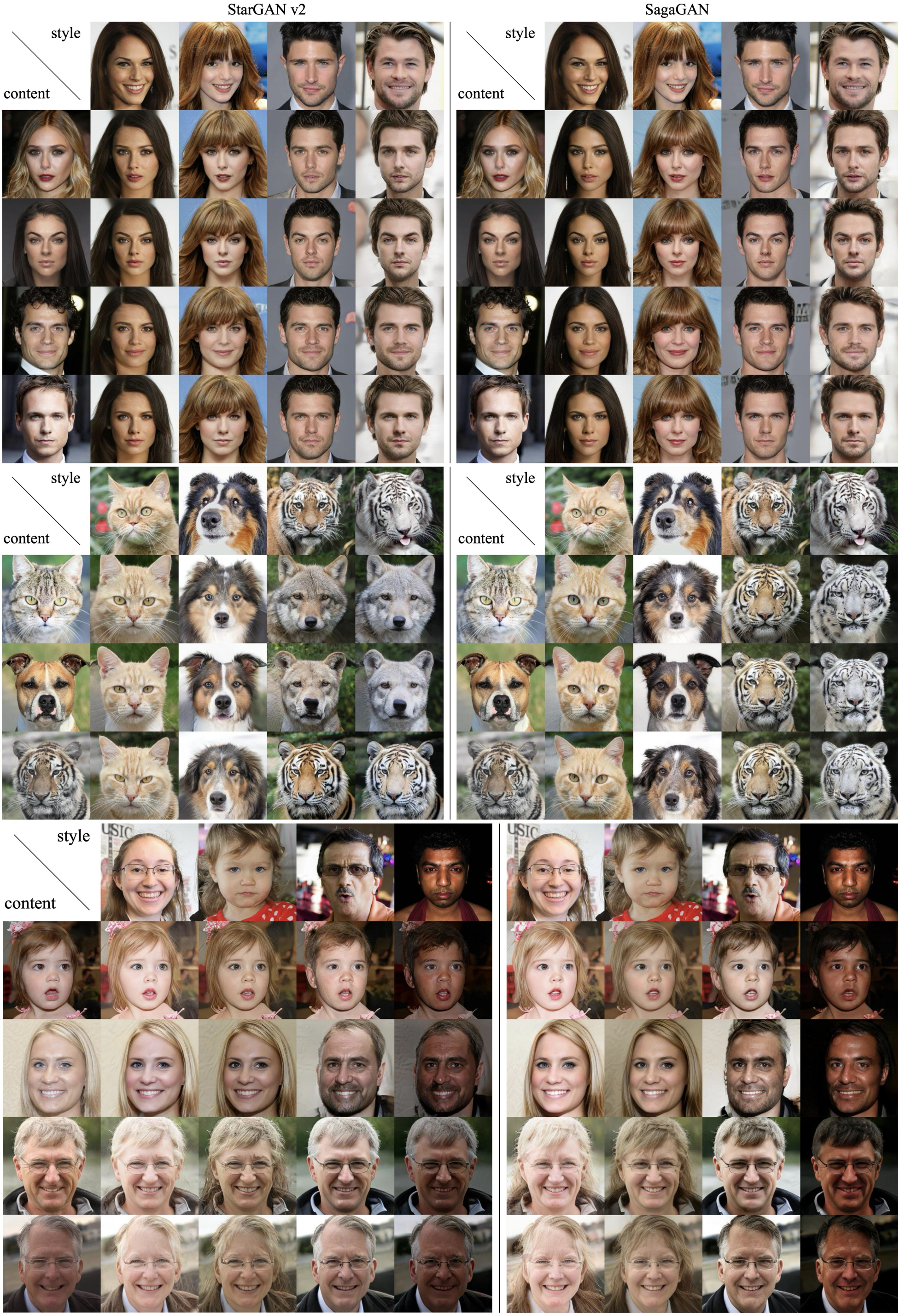

Figure 3: Qualitative comparison of image-to-image translation results between StarGAN v2 and SagaGAN on the CelebA-HQ, AFHQ, and FFHQ datasets. SagaGAN generates images with noticeably improved style transfer compared to the baseline StarGAN v2 model.

5 Conclusion

SagaGAN represents a significant advancement in the domain of image-to-image translation, particularly in the context of style transfer. By integrating the Gram matrix with AdaIN, SagaGAN surpasses the limitations of previous models by capturing more intricate style details and achieving a more nuanced transfer of visual styles. The novel G1 and G2 losses introduced in this study ensure a deeper and more precise adherence to the style characteristics of target images, contributing to the generation of images that not only exhibit higher quality but also greater diversity, as evidenced by our experimental results. As we progress, further exploration into optimizing the model's architecture and loss functions may yield even more refined results.

Acknowledgments

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University))

References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2019. URL https://api.semanticscholar.org/CorpusID:208617800.

[2] Clova AI et al. Stargan v2. https://github.com/clovaai/stargan-v2/blob/master/README.md#animal-faces-hq-dataset-afhq, 2024.

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *ArXiv*, abs/1508.06576, 2015. URL https://api.semanticscholar.org/CorpusID:13914930.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. URL https://api.semanticscholar.org/CorpusID:206593710.

[5] 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 *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:326772.

[6] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017. URL https://api.semanticscholar.org/CorpusID:6576859.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 5967–5976, 2016. URL https://api.semanticscholar.org/CorpusID:6200260.

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ArXiv*, abs/1603.08155, 2016. URL https://api.semanticscholar.org/CorpusID:980236.

[9] Béla Julesz. Visual pattern discrimination. *IRE Trans. Inf. Theory*, 8:84–92, 1962. URL https://api.semanticscholar.org/CorpusID:29648250.

[10] Béla Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91–97, 1981. URL https://api.semanticscholar.org/CorpusID:4327694.

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018. URL https://api.semanticscholar.org/CorpusID:54482423.

[12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2019. URL https://api.semanticscholar.org/CorpusID:209202273.

[13] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:34869018.

[14] Ming-Yu Liu, Thomas M. Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:3783306.

[15] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10550–10559, 2019. URL https://api.semanticscholar.org/CorpusID:146120584.

[16] NVIDIA Corporation. Flickr-faces-hq dataset (ffhq). https://github.com/NVlabs/ffhq-dataset, Year of Last Update.

[17] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019. URL https://api.semanticscholar.org/CorpusID:81981856.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL https://api.semanticscholar.org/CorpusID:14124313.

[19] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *ArXiv*, abs/1603.03417, 2016. URL https://api.semanticscholar.org/CorpusID:16728483.

[20] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. URL https://api.semanticscholar.org/CorpusID:4766599.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. URL https://api.semanticscholar.org/CorpusID:206770979.

[22] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:19046372.