Diagonal Attention and Style-based GAN for Content-Style Disentanglement in Image Generation and Translation

Gihyun Kwon[1] Jong Chul Ye[1,2]
Department of Bio and Brain Engineering[1], Graduate School of AI[2], KAIST
{cyclomon,jong.ye}@kaist.ac.kr

Figure 1: Full-resolution images synthesized with different diagonal attention (DAT) content codes and AdaIN style codes. (Left) Generated 1024×1024 images by our method trained using CelebA-HQ. (Right) Generated 512×512 images by our method trained using AFHQ. (a) A source image generated from arbitrary content and style codes. Samples generated by (b) varying style codes with fixed content codes, (c) varying content codes with fixed style codes, and (d) varying both codes. We can see that content code controls the spatial content such as directions, whereas style codes affects the styles like gender, hair color, etc.

Abstract

One of the important research topics in image generative models is to disentangle the spatial contents and styles for their separate control. Although StyleGAN can generate content feature vectors from random noises, the resulting spatial content control is primarily intended for minor spatial variations, and the disentanglement of global content and styles is by no means complete. Inspired by a mathematical understanding of normalization and attention, here we present a novel hierarchical adaptive Diagonal spatial ATtention (DAT) layers to separately manipulate the spatial contents from styles in a hierarchical manner. Using DAT and AdaIN, our method enables coarse-to-fine level disentanglement of spatial contents and styles. In addition, our generator can be easily integrated into the GAN inversion framework so that the content and style of translated images from multi-domain image translation tasks can be flexibly controlled. By using various datasets, we confirm that the proposed method not only outperforms the existing models in disentanglement scores, but also provides more flexible control over spatial features in the generated images.

1. Introduction

Recent development of Generative Adversarial Networks (GAN) [11] has enabled the generation of high-quality images that are indistinguishable to the human eye.

Despite its high performance, disentangling the attributes of the generated images is still an open problem.

For example, the content and style disentanglement is an important issue in many image generation tasks such as faces. Here, contents refer to the spatial information such as face direction, expression, whereas styles are related with other features such as color, makeup, gender. StyleGAN [18], which shows the state-of-the-art performance in image generation, tries to disentangle the style and content using the AdaIN codes [15] and the content feature vectors from random per-pixel noises, respectively. The AdaIN layer then combines the style and the content features to generate more realistic features at each resolution (see Fig. 2(a)). However, the content control by per-pixel noises is mostly for minor spatial variations so that the disentanglement of global contents and styles is by no means complete.

Recently, generative models that simultaneously use AdaIN and independent content latent codes [19, 1] have shown good performance in separating global style and content information. For example, in recent structured noise injection (SNI) approach [1], the latent code for content is generated by an additional neural network, which is used as an input tensor of the image generator composed of subsequent layers for style control using AdaIN (see Fig. 2(b)). Although SNI showed good performance in disentanglement, one of the major drawbacks is that the size of the input tensor is limited to relatively small resolution (e.g. 4×4). Therefore, the intended content control often fails to work properly due to the limited capacity.

To address these issues, here we introduce a novel Diagonal spatial ATtention (DAT) module to manipulate the content feature in a hierarchical manner. Specifically, the content code is applied to multiple layer features as diagonal attention maps at various resolutions as shown in Fig. 2(c). Despite the simplicity of diagonal attention, one of the important advantages of DAT is that the image content and style can be modulated independently in a symmetric manner; and similar to AdaIN for the styles, DAT enables the hierarchical control of the spatial content. These lead to an effective disentanglement of the content and style components in generated images

In addition, our method can be easily integrated into the state-of-the-art GAN inversion [42], allowing much more flexible post-hoc control of the content and style in the translated images from the multi-domain image translation.

2. Related works

2.1. Spatial attention

Spatial attention helps the visual learning tasks by highlighting the specific regions that contain important information. Several methods have used spatial attention to improve performance on some visual tasks: object detection [35, 40], semantic segmentation [9, 25], image captioning [2], and so on. Spatial attention has been further extended to non-traditional image generation tasks. For example, self-attention GAN [39] enhances the generation of geometrical and structural patterns. In the image-to-image translation tasks, recent methods achieved realistic generation performance with attention maps that focus on spatial areas of targeted objects [27, 8] or face components [38, 13, 4].SPADE [30] hierarchically applied spatial AdaIN to deliver the information of input condition map, which is, however, different from our model that disentangles style and content by using AdaIN and attention independently.

2.2. Disentangled representation

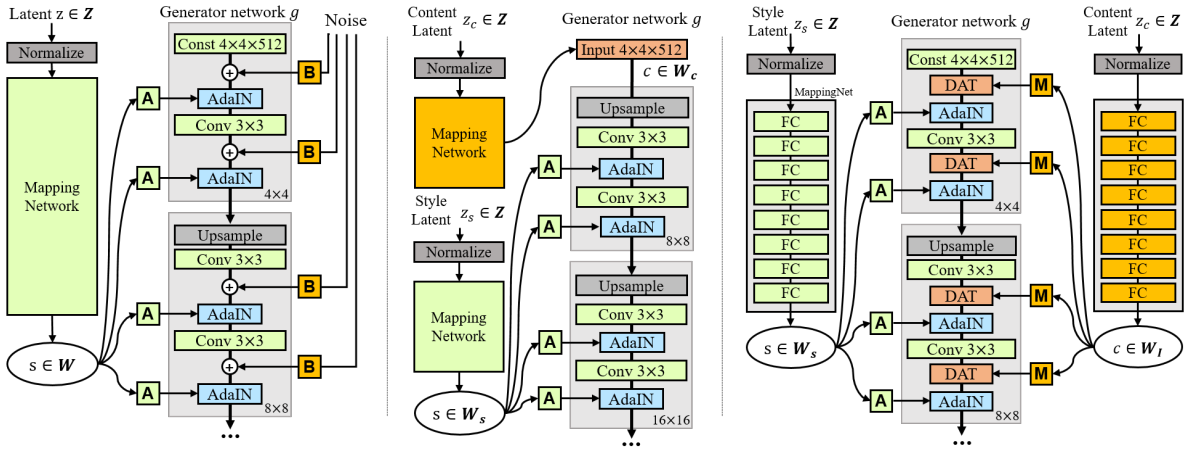
For disentangled image generation, several approaches have been proposed. Direct approaches rely on increasing the connection between the latent and image spaces [3, 23], a specialized training to constrain the latent space [24, 7], manipulating the prior distribution of latent [26], or using external attribute information [33]. Other approaches for disentanglement rely on the hierarchical structure of networks using layer-dependent latent variables in VAE to encode the disentangled attributes [34, 41, 22], using a tree-like latent variable structure [16], or synthesizing image components in several stages [32]. Despite the theoretical motivations, the above methods often suffer from poor generation quality due to limited network capacity or from disadvantages due to the need for additional attribute labels.

Recently, several authors propose to use an additional latent vector which controls independent attributes from the original one. For example, SC-GAN [19] separates style and content information using AdaIN along with input content codes. Additionally, there are methods which employ style-content disentanglement to improve the style transfer [20], and image translation [10]. Recently, a state-of-the-art style-content disentanglement was proposed in [1], which allows to control various spatial attributes by injecting structured noise as an input tensor of StyleGAN. Specifically, as shown in Fig. 2(b), a content latent z_c is processed by a specific neural network and directly used as an input tensor of the generator network. However, one of the drawbacks of this approach is its asymmetric architecture: although the style can be manipulated in a multi-resolution manner using hierarchical AdaIN layers, the content is controlled using a single input.

2.3. Our contributions

The architecture of our method, which we call the diagonal GAN, is shown in Fig. 2(c) and has several advantages over existing disentanglement methods.

- In contrast to the original StyleGAN in Fig. 2(a), the content and style code generation is symmetric by using similar code generators. Similar to the AdaIN



(a) StyleGAN (b) Structured Noise Injection (SNI) (c) Ours

Figure 2: Various style and content disentanglements: (a) StyleGAN with style and content control by AdaIN and per-pixel noises, respectively. (b) Structured Noise Injection (SNI) with additional content codes as an input tensor for a generator network. (c) Our approach with diagonal attention (DAT) and AdaIN for the content and style disentanglement.

layer, diagonal attention layer (DAT) enables the spatial control of the content in a hierarchical way that is difficult by SNI in Fig. 2(b).

- Although existing attention approach is implemented by multiplying a fully populated attention matrix, our approach is unique in that it uses a diagonal attention matrix to manipulate content information. While using a simple network architecture, this is a more efficient method as it enables much more powerful control of global content compared to the baseline StyleGAN model (see Fig. 2(a)).

3. Theory

3.1. Mathematics of Normalization and Attention

To understand the motivation of the proposed DAT layer, here we provide some mathematical analysis of existing normalization and attention modules in neural networks. Our analysis shows that the normalization and spatial attention have similar structure that can be exploited for style and content disentanglement.

Specifically, let H, W and C denote the height and width of the feature map, and the number of the feature channels, respectively. Then, for a given feature map $\boldsymbol{X} \in \mathbb{R}^{HW \times C}$, the AdaIN normalization layer output $\boldsymbol{Y} \in \mathbb{R}^{HW \times C}$ can be represented as follows:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{T} + \boldsymbol{R} \tag{1}$$

where the channel-directional transform $\boldsymbol{T}$ and the bias $\boldsymbol{R}$ are learned from the statistics of the feature maps. Specifically, $\boldsymbol{T}$ is a diagonal matrix that is calculated as the ratio of the standard deviations of the channel-wise input and target features, and $\boldsymbol{R}$ is the bias term that converts the input mean to the target mean.

Similarly, the spatial attention can be represented by

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{HW \times HW}$ is a fully populated matrix that is calculated from its own feature for the case of self-attention, or with the help of other domain features for the case of cross-domain attention. Since the transformation matrix $\boldsymbol{A}$ is applied to pixel-wise direction to manipulate the feature values of a specific location, it can control the spatial information such as shape and location.

In styleGAN, the content code $\boldsymbol{C}$, which is generated from per-pixel noises via the scaling network $\boldsymbol{B}$ (see Fig. 2(a)), is added to the feature $\boldsymbol{X}$ before the AdaIN layer. This leads to the following feature transform:

$$\boldsymbol{Y} = (\boldsymbol{X} + \boldsymbol{C})\boldsymbol{T} + \boldsymbol{R} = \boldsymbol{X}\boldsymbol{T} + \boldsymbol{R} + \boldsymbol{C}\boldsymbol{T} \tag{3}$$

Accordingly, the last term, $\boldsymbol{C}\boldsymbol{T}$, works as an additional bias term, which is different from the spatial attention that is multiplicative to the feature $\boldsymbol{X}$ (see (2)). Although one could potentially generate $\boldsymbol{C}$ so that the net effect is similar to $\boldsymbol{A}\boldsymbol{X}$, this would require a complicated content code generation network. This explains the fundamental limitation of the content control in the original styleGAN.

3.2. Diagonal Attention (DAT)

If normalization and attention in (1) and (2) are applied together, the output feature can be represented by

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}\boldsymbol{T} + \boldsymbol{R} \tag{4}$$

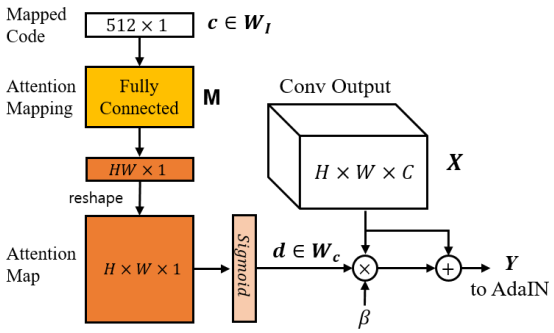


Figure 3: Generation of differential attention. With the attention mapping network M, the content code c is converted into an attention map. The map is multiplied element-wise to the convolution features. Since the attention maps of each layer contribute to the content information, they work as independent codes in the space $\boldsymbol{W}_c$.

One of the most important observations in this paper is that the combined equation (4) is the key for systematic style-content disentanglement. Specifically, $\boldsymbol{T}$ in (4) from AdaIN layer is a diagonal matrix obtained from a style code generator. Mathematically, the diagonal matrix $\boldsymbol{T}$ control the *row space* of the feature $\boldsymbol{X}$, which turns out to be the style control. Accordingly, we conjecture that the spatial content can be controlled by manipulating the remaining factor: the *column space* of the feature $\boldsymbol{X}$. Mathematically, this can be easily implemented by (4) using a diagonal attention matrix $\boldsymbol{A}$ that is obtained from another content code generator. The diagonal attention and diagonal normalization are then complimentary to each other, which are applied to different axes of the feature tensor to simultaneously control the two independent factors of the feature tensor $\boldsymbol{X}$. Furthermore, due to the symmetric role of AdaIN and DAT, they can be applied at each layer in a hierarchical manner as shown in Fig. 2(c).

Specifically, Fig. 2(c) describes the overall architecture of our proposed model. We adopt a method of using two different latent codes. In addition to the style latent code z_s, we use an independent latent code z_c to control the content information. More specifically, our style code z_s is mapped into a linearly distributed space $\boldsymbol{W}_s$ by several MLPs. Then the mapped code s is transformed into the parameters that can be applied to multiple layers as AdaIN. Similar to the style code mapping, the content code z_c is also mapped to a linear space $\boldsymbol{W}_I$ through a mapping function consisting of a series of MLPs. The mapped intermediate content code c can change the spatial information of the convolution output through the proposed attention mapping.

Figure 3 is a detailed diagram of our attention mapping network. Here, rather than directly estimating the diagonal component of $\boldsymbol{A}$, we are interested in estimating the perturbation from the identity attention. Specifically, the mapped content code c is converted into a vector with $HW \times 1$ dimension. Then the vector is reshaped into a differential attention map $\boldsymbol{d} \in \boldsymbol{W}_c$ which has the same spatial dimension $H \times W$ to that of the output from convolution layer. In order to avoid the undesired artifacts from excessive diversity in the attention map, we limit the value range of the differential attention with the help of sigmoid activation. Thanks to the diagonal attention map, the network output is then element-wise multiplied with feature map at each channel, which is added to the original feature map. In this stage, we use an additional parameter β, allowing the attention map of the network to learn the layer-wise contribution of content control. Since the contribution of attention can be calibrated by β depending on whether the layer is responsible for minor or major changes, an artifact from overemphasizing minor changes can be prevented.

Accordingly, the resulting feature output can be represented by

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \beta \boldsymbol{d} \odot \boldsymbol{x}_i = (\boldsymbol{I} + \beta \, \mathrm{diag}(\boldsymbol{d}))\, \boldsymbol{x}_i \qquad (5)$$

where $\mathrm{diag}(\boldsymbol{d})$ denotes the diagonal matrix whose diagonal elements is $\boldsymbol{d}$. This suggests that the resulting diagonal attention matrix is

$$\boldsymbol{A} = (\boldsymbol{I} + \beta \, \mathrm{diag}(\boldsymbol{d})) \ .$$

The DAT layer can also easily incorporate the per-pixel noises used in StyleGAN. However, care should be taken as per-pixel noise is only additive so that it can change minor spatial variations, whereas our diagonal attention is multiplicative so that we can control global spatial variations.

4. Method

4.1. Loss function

Our implementation is inspired by the original StyleGAN paper and the source code[1] implemented on PyTorch. Similar to StyleGAN, we choose the non-saturating Softplus loss with R_1 regularization for adversarial loss [28]. Specifically, softplus is formulated as $f(t) = \log(1 + \exp(t))$. Therefore, our adversarial losses are expressed as:

$$L_G = f(-D(X_{fake})), \quad X_{fake} = G(z_s, z_c)$$

$$L_D = f(D(X_{fake}) + f(-D(X_{real})) + \frac{\gamma}{2}\mathbb{E}[||\nabla D(X_{real})||_2^2]$$

where the last term of discriminator loss L_D is R_1 regularization. We also use Diversity-Sensitive (DS) loss [37] to encourage the attention network to yield diverse maps.

More specifically, if we sample two content codes z_c^1, z_c^2 and a style code z_s, our DS loss is defined as:

$$L_{ds} = \max\left(\lambda - \|G(z_s, z_c^1) - G(z_s, z_c^2)\|_1, 0\right)$$

[1] https://github.com/rosinality/style-based-gan-pytorch

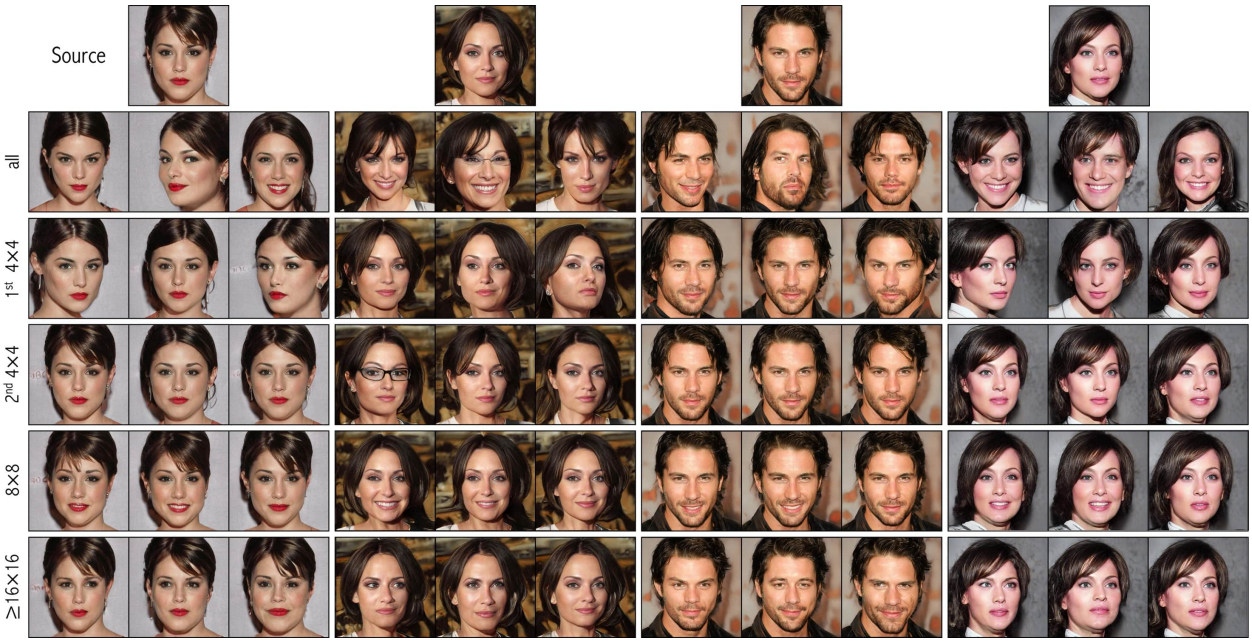


Figure 4: Our 1024×1024 full-resolution examples with fixed style and different content codes. The source images on top row are sampled from arbitrary DAT content and AdaIN style codes. The images on the second row are generated with changing content codes at entire layers under the fixed style code. The images in the following rows are sampled with changing the content codes at specific layers while fixing those of other layers. The hierarchical DAT layer can selectively control the extent of attribute changes.

where $G(z_s, z_c)$ is our generator with respect to the style code z_s and the content coder z_c, respectively. The objective of our DS loss is to maximize the L_1 distance between the generated images from different content codes with same style. However, directly optimizing the negative L_1 loss will lead to the explosion of the loss value. Therefore, we penalize DS loss with threshold λ so that the distance will not exceed λ. Accordingly, our total generator loss function is described as: $L_{G_{total}} = L_G + L_{ds}$

4.2. Experiment Settings

For qualitative evaluation, we report the results from the model trained on 1024×1024 CelebA-HQ [17] and 512×512 AFHQ [5]. In Supplementary Material, we also provide experimental results using flowers [29], birds [36], cars [21] data sets. Considering the number of parameters for attention mapping at high resolutions, we include the DAT layers up to the resolution of 256×256.

For quantitative evaluation, we compared our method with the baseline models that use input noises for content control. Among several methods using this approach, we use the state-of-the-art SNI [1] as a representative method. For fair comparison, we also included SNI trained with content DS loss as a baseline. We also use original StyleGAN results with per-pixel noises as another comparative model. For comparative studies with various parameter settings, we trained the models at the reduced resolution of 256×256 using 500K iterations (total of ∼4.7M samples). As baseline SNI presented results on models with and without adding per-pixel noises, we showed our results on both conditions. When training our models, we set the parameter λ in the DS loss as 0.3, as it showed the best performance. For more experimental settings, see Supplementary Materials.

For quantitative metrics, we use FID [14] for measuring the image quality and Perceptual Path Length (PPL) for measuring the disentanglement. PPL was first proposed in StyleGAN [18] to measure the perceptual distance between output images obtained with slightly changing the interpolated codes. A low PPL value means better disentanglement, since there is little interference of irrelevant features between two latent points. This can be also interpreted to mean that the latent space follows the linear trend. To measure the performance of the mapping networks that map both style and content code into their respective linear spaces, we compare the disentanglement performance by the PPL in the W (i.e. W_s and W_c) space.

5. Experimental Results

5.1. Qualitative Evaluation

Content and Style Disentanglement: Fig. 1 illustrates full-resolution images synthesized with different DAT and AdaIN codes. The left panel shows generated 1024×1024

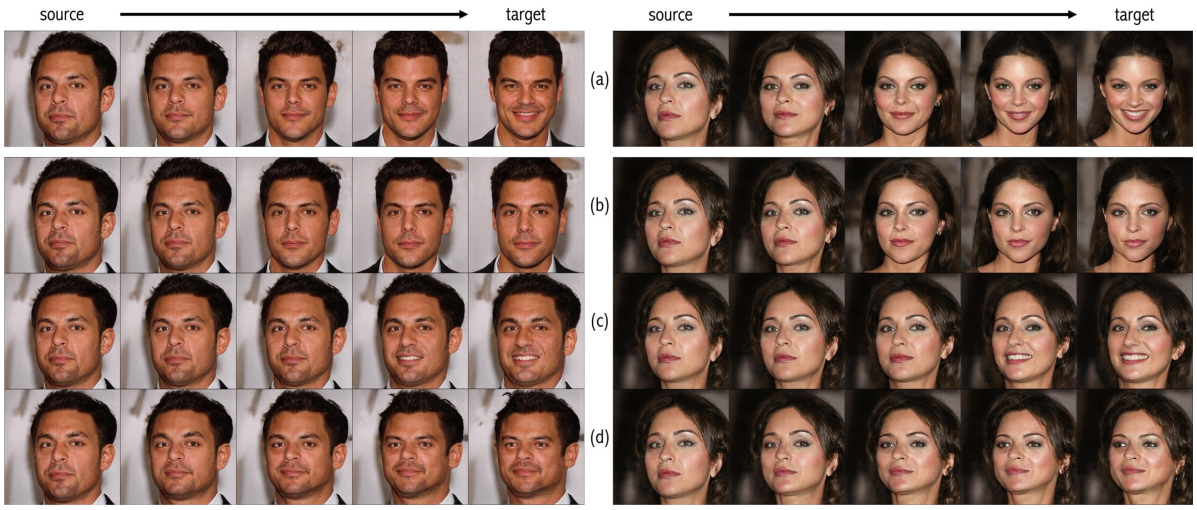


Figure 5: CelebA-HQ results from interpolated content codes. The images at each column are sampled from the same (interpolated) content code. (a) Results with interpolating content codes of all layers. (b) Interpolating 4×4 layer content codes. (c) Interpolating 8×8 layer content codes. (d) Interpolating content codes after 16×16 layer.

images by our method trained using CelebA-HQ, whereas the right panel shows generated 512×512 images by our method trained using AFHQ. For a given source image (a), which is generated from arbitrary DAT content and AdaIN style codes, the images in (b) show the generated samples with varying style codes and the fixed content code, whereas (c) illustrates samples with varying content codes and the fixed style codes. We can clearly see the effect of content code: the content of faces, such as the direction and components, vary. This is different from the effect of style codes in (b), which changes the hair color, gender, etc., while the face direction and components are fixed. By using specific style and content codes in (b)(c), the images in (c) shows that the face direction and components follow the content in (b), whereas the hair color, gender, etc are controlled by the style in (b). This experiments clearly indicates the powerful content and style disentanglement by our method. Although it is difficult to perfectly distinguish style and content, as both component contribute to face identity, the results show that our DAT provides disentangled content control, which can change the specific component with maintaining the identity much better compared to existing methods.

Hierarchical Content Disentanglement: We also show the hierarchical disentanglement ability by controlling diagonal attention map at each layer. The generated samples are presented in Figure 4. The source images on top row are sampled from arbitrary DAT content and AdaIN style codes. The images on the second row are generated with changing entire content code under the fixed style codes. We can observe the variations of entire spatial attributes including shape, rotation and facial expressions with consistent styles. The images in the following rows are sampled with changing the content codes at specific layers while fixing those of other layers. The first DAT at 4×4 layer mainly focuses on geometrical change, and the second 4×4 DAT changes hairstyles and eyes accessories. The 8×8 DAT layer mainly changes the lower part of facial expressions, and the DAT layers at higher resolutions give relatively minor variations such as hair curls and eyes. Quantitatively, our CelebA-HQ model showed satisfying performance of 7.32 in FID, compared with 5.17 of original StyleGAN.

Hierarchical Latent Interpolation: Figure 5 show the generated examples by interpolating DAT content codes $c \in W_I$ between two randomly sampled points with fixed style. The first row shows results from interpolating content codes of all layers, whereas the rest of the rows illustrate the results by interpolating specific layer content codes. Although similar latent interpolation in the first row (Figure 5(a)) could be done by StyleGAN, the fine spatial detail interpolation in Figure 5(b)-(d), such as mouth expressions, is not possible in StyleGAN. On the other hand, our method allows hierarchical content interpolation by interpolating the specific layer content codes. This hierarchical disentanglement can be also seen in our AFHQ results. For additional interpolation results using AFHQ and other data sets, see Supplementary Materials.

Direct Manipulation of Diagonal Attention: In order to verify the meaning of our diagonal attention maps, Figure 6 shows the generated samples by directly manipulating the diagonal attention maps at specific layers. With 4×4 maps, we can generate the faces with arbitrary direction by changing the activated regions. Also for 8×8 maps, we can control the mouth expression with high values on larger mouth areas. In 16×16 maps, we can control the size of eyes by manipulating activated pixel areas of eyes. In contrast to other style code based editing methods [31, 12, 6], our diagonal attention maps are shown to have a clear and intuitive relationship to different spatial regions. More examples and comparison results with other editing methods can be found in our Supplementary Materials.

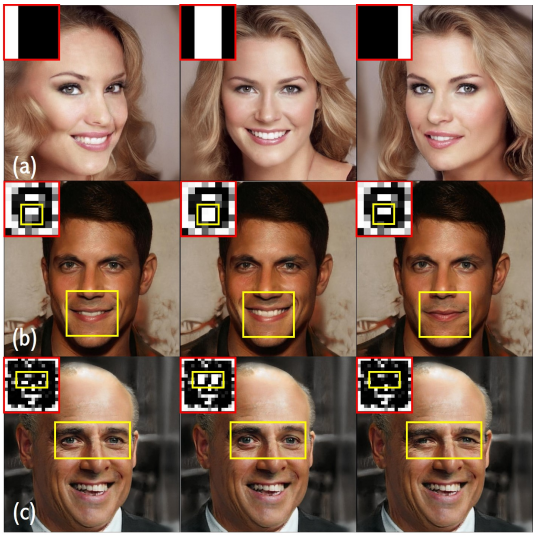


Figure 6: Direct attention map manipulation. By controlling the specific areas of attention, we can selectively change the facial attributes. Results from changing (a) the first 4×4 attention map, (b) the 2nd 8×8 attention map, and (c) the second 16×16 attention map. Yellow boxes represent the edited areas.

5.2. Quantitative Comparison Results

In Table 1, our model shows better performance in terms of disentanglement metric for almost all of the settings. Specifically, when we compare the models trained with both conditions of with and without per-pixel noises, we can see that our models show improved disentanglement metrics compared to SNI. The results clearly indicate that our diagonal attention map can obtain better disentanglement with rich control of the content than SNI. Even with the baseline SNI trained with DS loss, the model still could not overcome the limitation of insufficient capacity as indicated by the higher PPL scores. For further comparison, we also measured the disentanglement of not only the entire W space, but also the style space W_s and the content space W_c each. In all cases, our model achieved improved disentanglement performance with lower PPL scores. In addition, our models show comparable FID scores in almost all experimental settings. Although there is a slight degradation in some cases, they are from the expected trade-off between the image quality and the disentanglement as stated in [1]. To support the quantitative results, qualitative comparison with other methods are provided in Supplementary Materials, in addition to extensive ablation studies.

5.3. Inverting Disentangled Model

To further highlight the advantage of our method, we additionally implement a GAN inversion framework in which the real images are encoded into latent spaces, from which various output images are generated by simply manipulating content and style codes. For realistic image reconstruction, we use the modified version of state-of-the-art inversion method IDinvert [42] to include both DAT and AdaIN.

Per-pixel Noise		W PPL	W_s PPL	W_c PPL	FID
CQ	StyleGAN	85.96	-	-	8.87
	SNI	58.21	35.35	29.74	10.79
	SNI+DS	57.63	20.35	31.83	12.10
	Ours	**48.12**	**18.61**	**24.19**	10.90
AQ	StyleGAN	97.83	-	-	12.93
	SNI	65.22	43.62	18.82	11.32
	SNI+DS	69.70	45.79	18.20	15.35
	Ours	**63.44**	**42.17**	**17.73**	11.73
w/o Per-pixel Noise		W PPL	W_s PPL	W_c PPL	FID
CQ	StyleGAN	112.23	-	-	9.59
	SNI	70.64	33.11	31.35	10.93
	SNI+DS	90.81	35.61	45.12	9.89
	Ours	**53.72**	**21.92**	**30.15**	11.40
AQ	StyleGAN	374.72	-	-	13.92
	SNI	127.99	64.49	62.41	11.91
	SNI+DS	143.22	80.30	50.16	13.52
	Ours	**73.51**	**38.67**	**26.83**	12.67

Table 1: Comparison of FID and PPL scores of models trained using CelebA-HQ and AFHQ datasets at 256×256 resolution. A lower PPL indicates better disentanglement, and a lower FID indicates higher image quality. CQ: CelebA-HQ , AQ: AFHQ, s: Style,c: Content.

Specifically, we first pretrained our Diagonal GAN with multi-domain styles. Then, as shown in Fig. 8, we train the style encoder SE which has a double-head structure so that sampled style codes from each head represent the specific domain style (e.g. male, female). Additionally, the content encoder CE is trained so that it can generate the content code. The generated style and contents codes are fed into the pre-trained Diagonal GAN through AdaIN and DAT. Then, we train the network to reconstruct realistic input images. For encoder and diagonal GAN network training, we use 28,000 CelebA-HQ images with 256×256 resolution, which are split in two domains of males and females. For testing, we use 2,000 (1000 male, 1000 female) images. Detailed training process is elaborated in our Supplementary Materials.

Fig. 7 shows the synthesis results from our inversion model. First, auto-encoding reconstruction results confirm that the network can successfully generate similar outputs as the input images. Then, Figs. 7(b) show the results by changing the style codes. We can change the global styles from the inputs. In Figs. 7(c), we show the results by varying the content codes at each resolution layers. Thanks to the DAT layers, compared to the existing image translation models, our model has much more flexibility by allowing hierarchically control of both content and styles in the generated images.

For further evaluation, in Table 2, we compared the performance with the state-of-the-art image translation model StarGANv2 [5]. Since the existing StarGANv2 can only change the style similar to Figure 7(b), we measured the

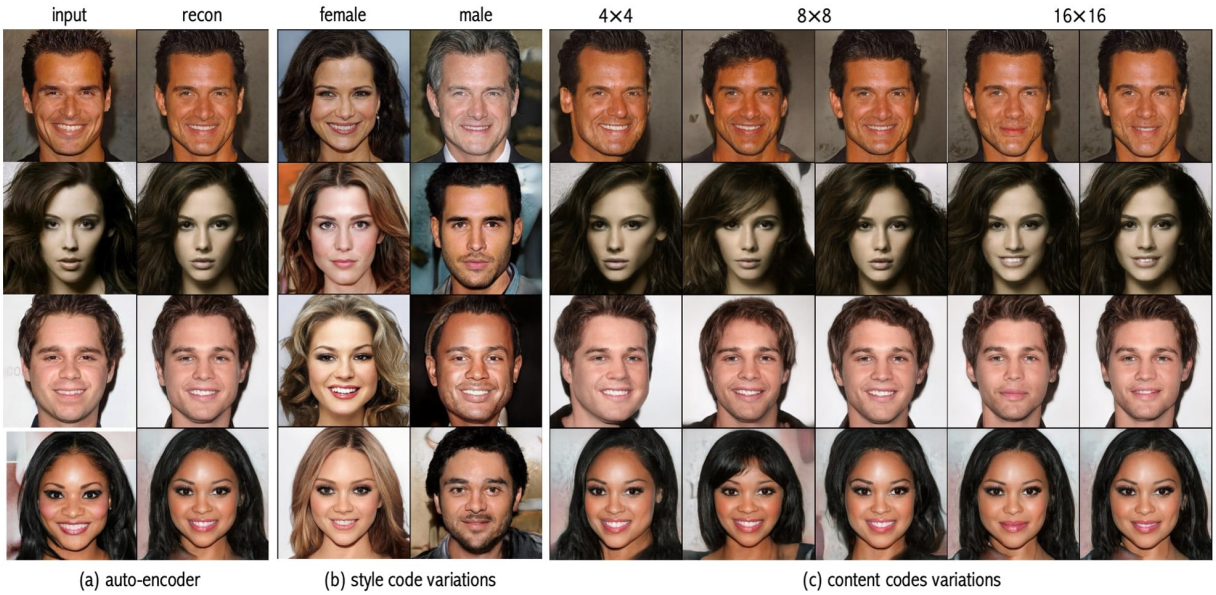


Figure 7: CelebA-HQ image synthesis results with our inversion model. (a) Auto-encoder reconstruction results from the input. (b) Images generated using random style codes by fixing content from the auto-encoder. We can specifically choose the domain (females or males) to translate the style. (c) With fixed style codes from auto-encoder and varying content codes, we can manipulate the content attributes. By changing the content codes of 4×4 layer, the face directional change. By changing the content codes of 8×8 layer, the hairstyle changes. By changing the content codes of 16×16 layer, the mouth expression varies.

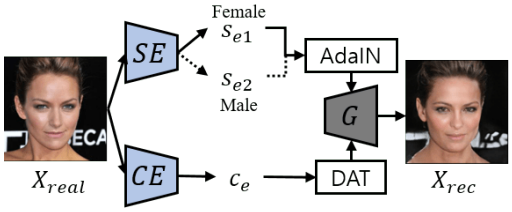


Figure 8: Network architecture for inverting pre-trained generator G. Our style encoder SE network produces multi-domain style codes. Content encoder CE produces domain-invariant content code.

Methods	Latent		Reference	
	FID	LPIPS	FID	LPIPS
StarGANv2	13.05	**0.453**	22.35	0.405
Ours	**11.12**	0.452	**18.11**	**0.407**

Table 2: Quantitative comparison of style synthesis using GAN inversion with CelebA-HQ. Lower FID means better image quality, and higher LPIPS means more diversity.

quantitative performance of style synthesis for fair comparison. Surprisingly, we achieved better image quality with comparable diversity even in style synthesis for both of latent-based sampling and reference-based transfer. The results show that our method has remarkable advantages, as it has a better image generation quality and more flexible content control than the existing state-of-the-art model. Detailed experiment settings and qualitative comparisons are provided in Supplementary Materials. To further verity the inversion performance of our proposed model, we also show comparison results of GAN inversion on different baselines in Supplementary Materials.

6. Conclusions

In this paper, we proposed a novel diagonal spatial attention (DAT) module as a complement to the AdaIN in order to disentangle the style and content information. The symmetric structure of DAT and AdaIN enabled the independent control of the style and content of features in a hierarchical manner. Our extensive experiments showed that the style and content attribute of images can be independently manipulated in a hierarchical manner, confirming the style and content disentanglement in high quality image generation. Moreover, the proposed method has also been successfully integrated into GAN inversion to achieve high quality image translation with better disentanglement of content and style.

Acknowledgement: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), and National Research Foundation of Korea (Grant NRF-2017M3C7A1047904).

References

[1] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5134–5142, 2020. 2, 5, 7

[2] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5659–5667, 2017. 2

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016. 2

[4] Ying-Cong Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I Pao, Jiaya Jia, et al. Semantic component decomposition for face attribute manipulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9859–9867, 2019. 2

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8188–8197, 2020. 5, 7

[6] E. Collins, R. Bala, B. Price, and S. Susstrunk. Editing in style: Uncovering the local semantics of gans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5770–5779, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 6

[7] Chris Donahue, Zachary C Lipton, Akshay Balsubramani, and Julian McAuley. Semantically decomposing the latent spaces of generative adversarial networks. *arXiv preprint arXiv:1705.07904*, 2017. 2

[8] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Chinnam. Spa-GAN: Spatial attention GAN for image-to-image translation. *IEEE Transactions on Multimedia*, 2020. 2

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3146–3154, 2019. 2

[10] Aviv Gabbay and Yedid Hoshen. Improving style-content disentanglement in image-to-image translation. *arXiv preprint arXiv:2007.04964*, 2020. 2

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press. 1

[12] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9841–9850. Curran Associates, Inc., 2020. 6

[13] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial attribute editing by only changing what you want. *IEEE Trans. Image Process.*, 28(11):5464–5478, 2019. 2

[14] 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 *Adv. Neural Inform. Process. Syst.*, pages 6626–6637, 2017. 5

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, pages 1501–1510, 2017. 2

[16] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. Generative adversarial image synthesis with decision tree latent controller. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6606–6615, 2018. 2

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Int. Conf. Learn. Represent.*, 2018. 5

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4401–4410, 2019. 2, 5

[19] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. pages 848–856. IEEE, 2019. 2

[20] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Int. Conf. Comput. Vis.*, pages 4422–4431, 2019. 2

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 5

[22] Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. *arXiv preprint arXiv:2002.10549*, 2020. 2

[23] Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. InfoGAN-CR: Disentangling generative adversarial networks with contrastive regularizers. *arXiv preprint arXiv:1906.06034*, 2019. 2

[24] Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *Adv. Neural Inform. Process. Syst.*, pages 2590–2599, 2018. 2

[25] Mengyu Liu and Hujun Yin. Cross attention network for semantic segmentation. In *IEEE Int. Conf. Image Process.*, pages 2434–2438. IEEE, 2019. 2

[26] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. pages 4402–4412, 2019. 2

[27] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Adv. Neural Inform. Process. Syst.*, pages 3693–3703, 2018. 2

[28] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*, 2018. 4

[29] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *In-*

dian Conference on Computer Vision, Graphics and Image Processing, Dec 2008. 5

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[31] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9240–9249, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 6

[32] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6490–6499, 2019. 2

[33] Nicki Skafte and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. In *Adv. Neural Inform. Process. Syst.*, pages 1018–1028, 2019. 2

[34] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016. 2

[35] Oytun Ulutan, ASM Iftekhar, and Bangalore S Manjunath. VSGNet: Spatial attention network for detecting human object interactions using graph convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13617–13626, 2020. 2

[36] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[37] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*, 2019. 4

[38] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *Eur. Conf. Comput. Vis.*, pages 417–432, 2018. 2

[39] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. pages 7354–7363. PMLR, 2019. 2

[40] Yunbo Zhang, Pengfei Yi, Dongsheng Zhou, Xin Yang, Deyun Yang, Qiang Zhang, and Xiaopeng Wei. CSANet: Channel and spatial mixed attention CNN for pedestrian detection. *IEEE Access*, 8:76243–76252, 2020. 2

[41] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from generative models. *arXiv preprint arXiv:1702.08396*, 2017. 2

[42] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2, 7