

# Attribute Manipulation Generative Adversarial Networks for Fashion Images

## -Supplementary Material-

### 1. Network Architecture

For the generator network choice in AMGAN, we use a structure similar to [3] and add an additional convolutional layer which outputs a single channel attention mask and consists of sigmoid activation function to arrange pixel values towards  $[0, 1]$  as shown in Table 1. The input of the generator is a tensor with "3+N+M" dimensions where  $N$  is the number of attribute values and  $M$  corresponds to the number of attributes. For the discriminator networks, we choose the PatchGAN architecture [2] similar to StarGAN [1]. The architecture of Discriminator  $D_I$  is shown in Table 2. For the Discriminator  $D_C$ , we removed the last 2 hidden layers and the input size is halved by 2 ( $h/2, w/2$ ). C: the number of output channels, K: kernel size, S: stride size, P: padding size, IN: instance normalization.

### 2. Additional Qualitative Results

We provide further qualitative results in Figure 1, 2, 3. For the DeepFashion dataset, the proposed AMGAN focuses mostly on the correct regions of clothes which contributes to its performance. The generated images from AMGAN does not "care" about the faces of wearers compared to StarGAN. For example, looking at images generated from StarGAN, while translating the input image, the face of the person gets more blurry since it does not consist of an attention mechanism

**Table 1:** Architecture of Generator Network  $G$

Part	Input $\rightarrow$ Output Shape	Layer Information
Down-sampling	$(h, w, 3 + N + M) \rightarrow (h, w, 64)$	CONV-(C64, K7x7, S1, P3), IN, ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(C128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(C256, K4x4, S2, P1), IN, ReLU
Bottleneck	$6 \times [(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)]$	6 Residual Blocks: CONV-(C256, K3x3, S1, P1), IN, ReLU
Upsampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	DECONV-(C128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 64)$	DECONV-(C64, K4x4, S2, P1), IN, ReLU
Generated Image	$(h, w, 64) \rightarrow (h, w, 3)$	CONV-(C3, K7x7, S1, P3), Tanh
Attention Mask	$(h, w, 64) \rightarrow (h, w, 1)$	CONV-(C1, K7x7, S1, P3), Sigmoid

**Table 2:** Architecture of Discriminator Network  $D_I$





















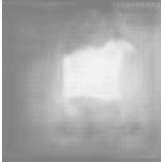



































Part	Input $\rightarrow$ Output Shape	Layer Information
Input Layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{h}{2}, 64)$	CONV-(C64, K4x4, S12, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{2}, \frac{h}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(C128, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{4}, \frac{h}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 128)$	CONV-(C256, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(C512, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(C1024, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(C2048, K4x4, S2, P1), Leaky ReLU
Output Adv. ( $D_{I_{src}}$ )	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(C1, K3x3, S1, P1)
Output Cls. ( $D_{I_{cls}}$ )	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, N)$	CONV-(C(N), K $\frac{h}{64}, \frac{w}{64}$ , S1, P0)

compared to the other methods. For sleeve attribute manipulation in Figure 2, AMGAN generates visually more satisfying images.







































































For the Shopping100k dataset, the most obvious difference can be seen from pattern and sleeve attribute manipulations. For instance, the generated images from AMGAN after sleeve attribute manipulation are more consistent with the input images in terms of color and pattern attributes.

## References

- [1] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, June 2018. [1](#)
- [2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. [1](#)
- [3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. [1](#)

Input	Attribute Manipulation	StarGAN	Ganimation	SaGAN	AMGAN	AMGAN: Generated Image	AMGAN: Attention Mask
	White Color						
	Black Color						
	Brown Color						
	Olive Color						
	Black Color						
	Purple Color						
	Red Color						
	Yellow Color						
	Orange Color						
	Red Color						

**Figure 1:** Attribute manipulation results on the DeepFashion for color attribute. The first two column show inputs and attribute manipulations while the other columns are images generated from each competing method. The last two columns are generated image ( $z$ ) and attention mask ( $\alpha$ ) outputs of AMGAN.

Input	Attribute Manipulation	StarGAN	Ganimation	SaGAN	AMGAN	AMGAN: Generated Image	AMGAN: Attention Mask
	Sleeveless						
	Sleeveless						
	Short Sleeve						
	Short Sleeve						
	Short Sleeve						
	Long Sleeve						
	Long Sleeve						
	Long Sleeve						
	Long Sleeve						
	Long Sleeve						

**Figure 2:** Attribute manipulation results on the DeepFashion for sleeve attribute. The first two column show inputs and attribute manipulations while the other columns are images generated from each competing method. The last two columns are generated image ( $z$ ) and attention mask ( $\alpha$ ) outputs of AMGAN.





**Figure 3:** Attribute manipulation results on the Shopping100k dataset. The first two column show inputs and attribute manipulations while the other columns are images generated from each competing method. The last two columns are generated image ( $z$ ) and attention mask ( $\alpha$ ) outputs of AMGAN.