

STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing

Ming Liu^{1*}, Yukang Ding², Min Xia¹, Xiao Liu², Errui Ding², Wangmeng Zuo^{(✉)1,3}, Shilei Wen²

¹Harbin Institute of Technology, ²Department of Computer Vision Technology (VIS), Baidu Inc., ³Peng Cheng Laboratory, Shenzhen

csmliu@outlook.com, csmxia@gmail.com, wmuo@hit.edu.cn,

{dingyukang, liuxiao12, dingerrui, wenshilei}@baidu.com

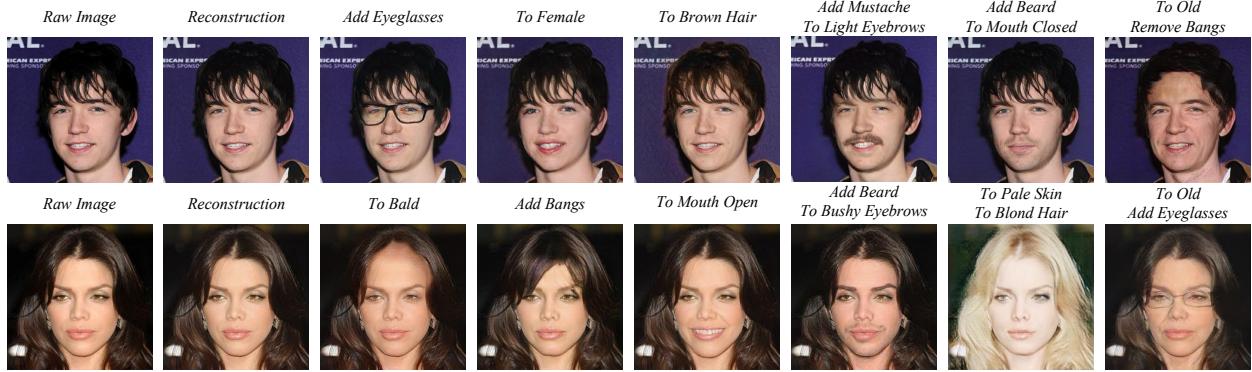


Figure 1: High resolution (384×384) results of STGAN for facial attribute editing, and more results are given in the suppl.

Abstract

Arbitrary attribute editing generally can be tackled by incorporating encoder-decoder and generative adversarial networks. However, the bottleneck layer in encoder-decoder usually gives rise to blurry and low quality editing result. And adding skip connections improves image quality at the cost of weakened attribute manipulation ability. Moreover, existing methods exploit target attribute vector to guide the flexible translation to desired target domain. In this work, we suggest to address these issues from selective transfer perspective. Considering that specific editing task is certainly only related to the changed attributes instead of all target attributes, our model selectively takes the difference between target and source attribute vectors as input. Furthermore, selective transfer units are incorporated with encoder-decoder to adaptively select and modify encoder feature for enhanced attribute editing. Experiments show that our method (i.e., STGAN) simultaneously improves attribute manipulation accuracy as well as perception quality, and performs favorably against state-of-the-arts in arbitrary facial attribute editing and season translation.

1. Introduction

Image attribute editing, aiming at manipulating an image to possess desired attributes, is an interesting but challenging problem with many real-world vision applications. On one hand, it is impracticable to collect paired images with and without desirable attributes (e.g., *female* and *male* face images of the same person). Thus, unsupervised generative learning models, e.g., generative adversarial networks (GANs) [9], have attracted upsurging attention in attribute editing. On the other hand, arbitrary attribute editing actually is a multi-domain image-to-image translation task. Learning single translation model for each specific attribute editing task may achieve limited success [20, 29, 34]. But it is ineffective in exploiting the entire training data, and the learned models grows exponentially along with the number of attributes. To handle this issue, several arbitrary attribute editing approaches [7, 11, 26] have been developed, which usually (i) use encoder-decoder architecture, and (ii) take both source image and target attribute vector as input.

Albeit their extensive deployment, encoder-decoder networks remain insufficient for high quality attribute editing. Attribute can be of either local, global, or abstract characteristic of the image. In order to properly manipulate image attribute, spatial pooling or downsampling generally are required to obtain high-level abstraction of image content

*Work done during an internship at Baidu.



Figure 2: Reconstruction results of AttGAN [11], StarGAN [7], and our STGAN.

and attributes. For example, auto-encoder architecture is adopted in [11, 17, 26], and shallow encoder-decoder with residual blocks is used in [7, 34]. However, the introduction of bottleneck layer, *i.e.*, the innermost feature map with minimal spatial size, gives rise to blurry and low quality editing result. As a remedy, some researchers suggest to add one [11] or multiple [14] skip connections between encoder and decoder layers. Unfortunately, as further shown in Sec. 3.1, the deployment of skip connections improves image quality of editing result but is harmful to attribute manipulation ability of learned model. Another possible solution is to employ spatial attention network to allow attribute-specific region editing [32], which, however, is effective only for local attributes and not designed for arbitrary attribute editing.

Moreover, most existing methods exploit both source image and target attribute vector for arbitrary attribute editing. In particular, the encoders in [11, 17] only take source image as input to produce latent code, and then the decoders utilize both latent code and target attribute vector to generate editing result. In contrast, StarGAN [7] directly takes source image and target attribute vector as input. However, for arbitrary attribute editing only the attributes to be changed are required, taking full target attribute vector as input may even have adverse effect on editing result. As shown in Fig. 2, although all attributes keep unchanged, unwanted changes and visual degradation can be observed in the results by AttGAN [11] and StarGAN [7], mainly ascribing to the limitation of encoder-decoder and the use of target attribute vector as input.

To address the above issues, this work investigates arbitrary attribute editing from selective transfer perspective and presents a STGAN model. In terms of *selective*, our STGAN is suggested to (i) only consider the attributes to be changed, and (ii) selectively concatenate encoder feature in editing attribute irrelevant regions with decoder feature. In terms of *transfer*, our STGAN is expected to adaptively modify encoder feature to match the requirement of varying editing task, thereby providing a unified model for handling both local and global attributes.

To this end, instead of full target attribute vector, our STGAN takes the difference between target and source at-

tribute vectors as input to encoder-decoder. Subsequently, selective transfer units (STUs) are proposed to adaptively select and modify encoder feature, which is further concatenated with decoder feature for enhancing both image quality and attribute manipulation ability. In particular, STU is added to each pair of encoder and decoder layers, and takes both encoder feature, inner state, and difference attribute vector into consideration for exploiting cross-layer consistency and task specificity. From Figs. 1 and 2, our STGAN can generate high quality and photo-realistic results for arbitrary attribute editing, and obtain near-ideal reconstruction when the target and source attributes are the same. To sum up, the contribution of this work involves:

- Instead of all target attributes, difference attribute vector is taken as input to enhance the flexible translation of attributes and ease the training procedure.
- Selective transfer units are presented and incorporated with encoder-decoder for simultaneously improving attribute manipulation ability and image quality.
- Experimental results show that our STGAN performs favorably against state-of-the-arts in arbitrary facial attribute editing and season translation.

2. Related Work

Encoder-Decoder Architecture. In their pioneer work [12], Hinton and Zemel proposed an autoencoder network, which consists of an encoder to map the input into *latent code* and a decoder to recover from the *latent code*. Subsequently, denoising autoencoders [30] are presented to learn representation robust to partial corruption. Kingma and Welling [16] suggested a Variational Autoencoder (VAE), which validates the feasibility of encoder-decoder architecture to generate unseen images. Recent studies show that skip connections [14, 28] between encoder and decoder layers usually benefit the training stability and visual quality of generated images. However, as discussed in Sec. 3.1, skip connections actually improves image quality at the cost of weakened attribute manipulation ability, and should be carefully used in arbitrary attribute editing.

Generative Adversarial Networks. GAN [9, 27] is originally proposed to generate images from random noise, and generally consists of a generator and a discriminator which are trained in an adversarial manner and suffer from the mode collapse problem. Recently, enormous efforts have been devoted to improving the stability of learning. In [3, 10], Wasserstein-1 distance and gradient penalty are suggested to improve stability of the optimization process. In [18], the VAE decoder and GAN generator are collapsed into one model and optimized by both reconstruction and adversarial loss. Conditional GAN (cGAN) [14, 24] takes conditional variable as input to the generator and discriminator to generate image with desired properties. As a re-



Figure 3: Results of AttGAN [11] variants for reconstructing input image. Please zoom in for better observation.

sult, GAN has become one of the most prominent models for versatile image generation [9, 27], translation [14, 34], restoration [19, 21] and editing [25] tasks.

Image-to-Image Translation. Image-to-image translation aims at learning cross-domain mapping in supervised or unsupervised settings. Isola *et al.* [14] presented a unified pix2pix framework for learning image-to-image translation from paired data. Improved network architectures, *e.g.*, cascaded refinement networks [4] and pix2pixHD [31], are then developed to improve the visual quality of synthesized images. As for unpaired image-to-image translation, additional constraints, *e.g.*, cycle consistency [34] and shared latent space [22], are suggested to alleviate the inherent ill-posedness of the task. Nonetheless, arbitrary attribute editing actually is a multi-domain image-to-image translation problem, and cannot be solved with scalability by aforementioned methods. To address this issue, [2] and [13] decouple generators by learning domain-specific encoders/decoders with shared latent space, but are still limited in scaling to change multiple attributes of an image.

Facial Attribute Editing. Facial attribute editing is an interesting multi-domain image-to-image translation problem and has received considerable recent attention. While several methods have been proposed to learn single translation model for each specific attribute editing task [5, 20, 29, 32], they suffer from the limitation of image-to-image translation and cannot well scale to arbitrary attribute editing. Therefore, researchers resort to learning a single model for arbitrary attribute editing. IcGAN [26] adopts an encoder to generate latent code of an image, and a cGAN to decode latent code conditioned on target attributes. However, IcGAN first trains the cGAN model followed by the encoders, greatly restricting its reconstruction ability. Lamplle *et al.* [17] trained the FaderNet in an end-to-end manner by imposing adversarial constraint to enforce the independence between latent code and attributes. ModularGAN [33] presents a feasible solution to connect specific attribute editing to arbitrary attribute editing, but its computation time gradually increases along with the number of attributes to be changed. StarGAN [7] and AttGAN [11] elaborately tackle arbitrary attribute editing by taking target attribute vector as input to the transform model. In this work, we analyze the limitation of StarGAN [7] and AttGAN [11], and further develop a STGAN for simultaneously enhancing the attribute manipulation ability and image quality.

Method	AttGAN-ED	AttGAN	AttGAN-2s	AttGAN-UNet
PSNR/SSIM	22.68/0.758	24.07/0.841	26.13/0.897	29.66/0.929

Table 1: Reconstruction evaluation of AttGAN [11] variants.

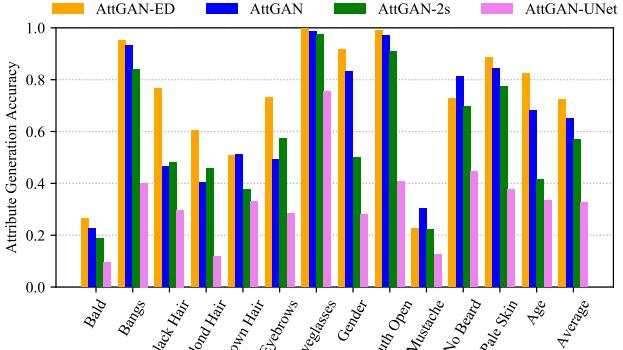


Figure 4: Attribute generation accuracy of AttGAN [11] variants.

3. Proposed Method

This section presents our proposed STGAN for arbitrary attribute editing. To begin with, we use AttGAN as an example to analyze the limitation of skip connections. Then, we formulate STGAN by taking difference attribute vector as input and incorporating selective transfer units into encoder-decoder structure. Finally, network architecture (see Fig. 5) and model objective of STGAN are provided.

3.1. Limitation of Skip Connections in AttGAN

StarGAN [7] and AttGAN [11] adopt encoder-decoder structure, where spatial pooling or downsampling are essential to obtain high level abstract representation for attribute manipulation. Unfortunately, downsampling irreversibly diminishes spatial resolution and fine details of feature map, which cannot be completely recovered by transposed convolutions and the results are prone to blurring or missing details. To enhance image quality of editing result, AttGAN [11] applies one skip connection between encoder and decoder, but we will show that it is still limited.

To analyze the effect and limitation of skip connections, we test four variants of AttGAN on the test set: (i) AttGAN w/o skip connection (AttGAN-ED), (ii) AttGAN model released by He *et al.* [11] with one skip connection (AttGAN), (iii) AttGAN with two skip connections (AttGAN-2s), and (iv) AttGAN with all symmetric skip connections [28] (AttGAN-UNet). Table 1 lists the PSNR/SSIM results of reconstruction by keeping target attribute vector the same as the source one, and Fig. 3 shows the reconstruction results of an image. It can be seen that adding skip connections does benefit the reconstruction of fine details, and better result can be obtained with the increase of skip connections. By setting target attribute vector different from source one, Fig. 4 further assesses the facial attribute generation accuracy via a facial attribute classification model¹. While

¹We train the model on CelebA [23] dataset which can achieve 94.5% mean accuracy on the 13 attributes we use.

adding one skip connection, *i.e.*, AttGAN, only slightly decreases generation accuracy for most attributes, notable degradation can be observed by adding multiple skip connections. Thus, the deployment of skip connections improves reconstruction image quality at the cost of weakened attribute manipulation ability, mainly attributing to that skip connection directly concatenates encoder and decoder features. To circumvent this dilemma, we present our STGAN to employ selective transfer units to adaptively transform encoder features guided by attributes to be changed.

3.2. Taking Difference Attribute Vector as Input

Both StarGAN [7] and AttGAN [11] take target attribution vector att_t and source image x as input to the generator. Actually, the use of full target attribution vector is redundant and may be harmful to editing result. In Fig. 2, the target attribution vector att_t is exactly the same as on the source one att_s , but StarGAN [7] and AttGAN [11] may manipulate some unchanged attributes by mistake. From Fig. 2, after editing the face image with blond hair becomes more blond. Moreover, they even incorrectly adjust hair length of a source image with the attribute *female*.

For arbitrary image attribute editing, instead of full target attribute vector, only the attributes to be changed should be considered to preserve more information of source image. So we define the difference attribute vector as the difference between target and source attribute vectors,

$$\text{att}_{\text{diff}} = \text{att}_t - \text{att}_s. \quad (1)$$

Taking att_{diff} as input can bring several distinctive merits. First, the attributes to be changed are only a small set of attribute vector, and the use of att_{diff} usually makes the model easier to train. Second, in comparison to att_t , att_{diff} can provide more valuable information for guiding image attribute editing, including whether an attribute is required to edit or not, toward what direction an attribute should be changed. The information can then be utilized to design proper model to transform and concatenate encoder feature with decoder feature, and improve image reconstruction quality without sacrifice of attribute manipulation accuracy. Finally, in practice att_{diff} actually is more convenient to be provided by user. When taking att_t as input, the user is required to either manually supply all target attributes, or modify source attributes provided by some attribute prediction method.

3.3. Selective Transfer Units

Fig. 5 shows the overall architecture of our STGAN. Instead of directly concatenating encoder with decoder features via skip connection, we present selective transfer unit (STU) to selectively transform encoder feature, making it compatible and complementary to decoder feature. Naturally, the transform is required to be adaptive to the changed

attributes, and be consistent among different encoder layers. Thus, we modify the structure of GRU [6, 8] to build STUs for passing information from inner layers to outer layers.

Without loss of generality, we use the l -th encoder layer as an example. Denote by $\mathbf{f}_{\text{enc}}^l$ the encoder feature of the l -th layer, and \mathbf{s}^{l+1} the hidden state from the $l+1$ -th layer. For convenience, the difference attribute vector att_{diff} is stretched to have the same spatial size of \mathbf{s}^{l+1} . Different from sequence modeling, feature maps across layers are of different spatial size. So we first use transposed convolution to upsample hidden state \mathbf{s}^{l+1} ,

$$\hat{\mathbf{s}}^{l+1} = \mathbf{W}_t *_{\mathcal{T}} [\mathbf{s}^{l+1}, \text{att}_{\text{diff}}], \quad (2)$$

where $[\cdot, \cdot]$ denotes the concatenation operation, and $*_{\mathcal{T}}$ denotes transposed convolution. Then, STU adopts the mathematical model of GRU to update the hidden state \mathbf{s}^l and transformed encoder feature \mathbf{f}_t^l ,

$$\mathbf{r}^l = \sigma(\mathbf{W}_r * [\mathbf{f}_{\text{enc}}^l, \hat{\mathbf{s}}^{l+1}]), \quad (3)$$

$$\mathbf{z}^l = \sigma(\mathbf{W}_z * [\mathbf{f}_{\text{enc}}^l, \hat{\mathbf{s}}^{l+1}]), \quad (4)$$

$$\mathbf{s}^l = \mathbf{r}^l \circ \hat{\mathbf{s}}^{l+1}, \quad (5)$$

$$\hat{\mathbf{f}}_t^l = \tanh(\mathbf{W}_h * [\mathbf{f}_{\text{enc}}^l, \mathbf{s}^l]), \quad (6)$$

$$\mathbf{f}_t^l = (1 - \mathbf{z}^l) \circ \hat{\mathbf{s}}^{l+1} + \mathbf{z}^l \circ \hat{\mathbf{f}}_t^l, \quad (7)$$

where $*$ denotes the convolution operation, \circ denotes entry-wise product, and $\sigma(\cdot)$ stands for the sigmoid function.

The introduction of the reset gate \mathbf{r}^l and update gate \mathbf{z}^l allows us control the contribution of hidden state, difference attribute vector, and encoder feature in a selective manner. Moreover, the convolution transform and linear interpolation in Eqns. (6) and (7) provide an adaptive means for the transfer of encoder feature and its combination with hidden state. In comparison to GRU where \mathbf{f}_t^l is adopted as the output of hidden state, we take \mathbf{s}^l as the output of hidden state and \mathbf{f}_t^l as the output of transformed encoder feature. And experiments empirically validate that such modification can bring moderate gains on attribute generation accuracy.

3.4. Network Architecture

Our STGAN is comprised of two components, *i.e.*, a generator G and a discriminator D . Fig. 5 illustrates the network structure of G consisting of an encoder G_{enc} for abstract latent representation and a decoder G_{dec} for target image generation. The encoder G_{enc} contains five convolution layers with kernel size 4 and stride 2, while the decoder G_{dec} has five transposed convolution layers. Besides, STU is applied right after each of the first four encoder layers, denoted by $(\mathbf{f}_t^l, \mathbf{s}^l) = G_{\text{st}}^l(\mathbf{f}_{\text{enc}}^l, \mathbf{s}^{l+1}, \text{att}_{\text{diff}})$.

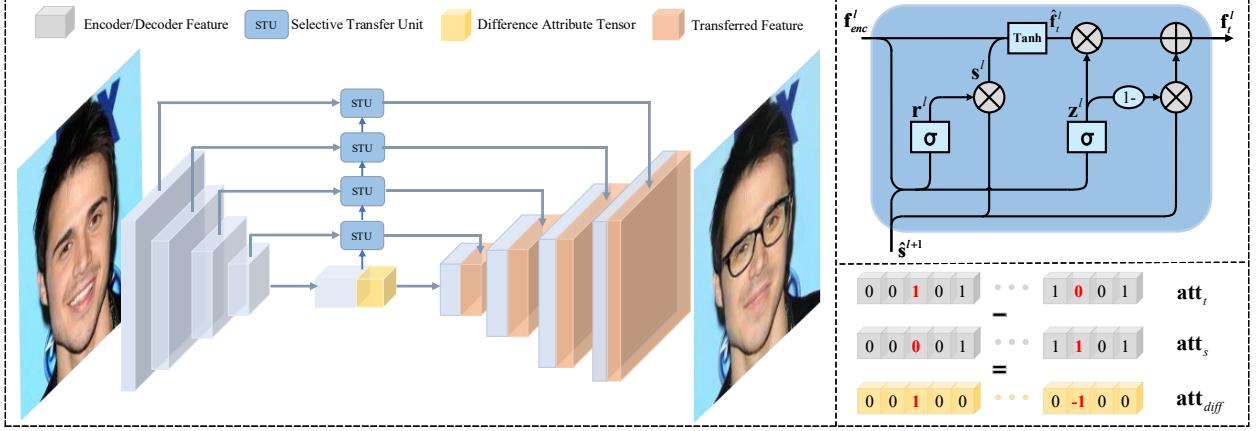


Figure 5: The overall structure of STGAN. On the left is the generator. The top-right figure shows detailed STU structure, and all variables marked in this figure share same dimension (e.g., 64×64). The difference attribute vector of adding *Eyeglasses* and removing *Mouth Open* attributes is shown on the bottom-right.

The discriminator D has two branches D_{adv} and D_{att} . D_{adv} consists of five convolution layers and two fully-connected layers to distinguish whether an image is a fake image or a real one. D_{att} shares the convolution layers with D_{adv} , but predicts an attribute vector by another two fully-connected layers. Please refer to the suppl. for more details on the network architecture.

3.5. Loss Functions

Given an input image \mathbf{x} , the encoder features can be obtained by,

$$\mathbf{f} = G_{enc}(\mathbf{x}), \quad (8)$$

where $\mathbf{f} = \{\mathbf{f}_{enc}^1, \dots, \mathbf{f}_{enc}^5\}$. Then, guided by \mathbf{att}_{diff} , STUs are deployed to transform encoder features for each layer,

$$(\mathbf{f}_t^l, \mathbf{s}^l) = G_{st}^l(\mathbf{f}_{enc}^l, \mathbf{s}^{l+1}, \mathbf{att}_{diff}), \quad (9)$$

Note that we adopt four STUs, and directly pass \mathbf{f}_{enc}^5 to G_{dec} . The STUs deployed in different layers do not share parameters due to that (i) the dimensions are different and (ii) the features of inner layers are more abstract than those of the outer layers.

Let $\mathbf{f}_t = \{\mathbf{f}_t^1, \dots, \mathbf{f}_t^4\}$. Thus, the editing result of G_{dec} can be given by,

$$\hat{\mathbf{y}} = G_{dec}(\mathbf{f}_{enc}^5, \mathbf{f}_t), \quad (10)$$

and can be written by,

$$\hat{\mathbf{y}} = G(\mathbf{x}, \mathbf{att}_{diff}). \quad (11)$$

In the following, we detail the reconstruction, adversarial, and attribute manipulation losses which are collaborated to train our STGAN.

Reconstruction loss. When the target attributes are exactly the same as source ones, i.e., $\mathbf{att}_{diff} = \mathbf{0}$, it is natural to require that the editing result approximates the source image. Thus the reconstruction loss is defined as,

$$\mathcal{L}_{rec} = \|\mathbf{x} - G(\mathbf{x}, \mathbf{0})\|_1, \quad (12)$$

where the ℓ_1 -norm $\|\cdot\|_1$ is adopted for preserving the sharpness of reconstruction result.

Adversarial loss. When the target attributes are different from source ones, i.e., $\mathbf{att}_{diff} \neq \mathbf{0}$, the ground-truth of editing result will be unavailable. Therefore, adversarial loss [9] is employed for constraining the editing result to be indistinguishable from real images. In particular, we follow Wasserstein GAN (WGAN) [3] and WGAN-GP [10], and define the losses for training D_{adv} and G as,

$$\max_{D_{adv}} \mathcal{L}_{D_{adv}} = \mathbb{E}_{\mathbf{x}} D_{adv}(\mathbf{x}) - \mathbb{E}_{\hat{\mathbf{y}}} D_{adv}(\hat{\mathbf{y}}) + \lambda \mathbb{E}_{\hat{\mathbf{x}}} [(\|\nabla_{\hat{\mathbf{x}}} D_{adv}(\hat{\mathbf{x}})\|_2 - 1)^2], \quad (13)$$

$$\max_G \mathcal{L}_{G_{adv}} = \mathbb{E}_{\mathbf{x}, \mathbf{att}_{diff}} D_{adv}(G(\mathbf{x}, \mathbf{att}_{diff})), \quad (14)$$

where $\hat{\mathbf{x}}$ is sampled along lines between pairs of real and generated images.

Attribute manipulation loss. Even the ground-truth is missing, we can require the editing result to possess the desired target attributes. Thus, we introduce an attribute classifier D_{att} which shares the convolution layers with D_{adv} , and define the following attribute manipulation losses for training D_{att} and generator G ,

$$\mathcal{L}_{D_{att}} = - \sum_{i=1}^c [\mathbf{att}_s^{(i)} \log D_{att}^{(i)}(\mathbf{x}) + (1 - \mathbf{att}_s^{(i)}) \log (1 - D_{att}^{(i)}(\mathbf{x}))], \quad (15)$$

$$\mathcal{L}_{G_{att}} = - \sum_{i=1}^c [\mathbf{att}_t^{(i)} \log D_{att}^{(i)}(\hat{\mathbf{y}}) + (1 - \mathbf{att}_t^{(i)}) \log (1 - D_{att}^{(i)}(\hat{\mathbf{y}}))], \quad (16)$$

where $\mathbf{att}_{s/t}^{(i)}$ ($D_{att}^{(i)}(\mathbf{x})$) denotes the i -th attribute value of $\mathbf{att}_{s/t}$ ($D_{att}(\mathbf{x})$).

Method	IcGAN	FaderNet	AttGAN	StarGAN	STGAN
PSNR/SSIM	15.28/0.430	30.62/0.908	24.07/0.841	22.80/0.819	31.67/0.948

Table 2: Reconstruction quality of the comparison methods on facial attribute editing task.

Model Objective. Taking the above losses into account, the objective to train the discriminator D can be formulated as,

$$\min_D \mathcal{L}_D = -\mathcal{L}_{D_{adv}} + \lambda_1 \mathcal{L}_{D_{att}}, \quad (17)$$

and that for the generator G is,

$$\min_G \mathcal{L}_G = -\mathcal{L}_{G_{adv}} + \lambda_2 \mathcal{L}_{G_{att}} + \lambda_3 \mathcal{L}_{rec}, \quad (18)$$

where λ_1 , λ_2 , and λ_3 are the model tradeoff parameters.

4. Experiments

We train the model by the ADAM [15] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is initialized as 2×10^{-4} and decays to 2×10^{-5} for fine-tuning after 100 epochs. In all experiments, the tradeoff parameters in Eqns. (17) and (18) are set to $\lambda_1 = 1$, $\lambda_2 = 10$ and $\lambda_3 = 100$. All the experiments are conducted in the TensorFlow [1] environment with cuDNN 7.1 running on a PC with Intel(R) Xeon(R) E3-1230v5 CPU 3.40GHz and Nvidia GTX1080Ti GPU. The source code can be found at <https://github.com/csmliu/STGAN.git>.

4.1. Facial Attribute Editing

Following [7, 11], we first evaluate our STGAN for arbitrary facial attribute editing on the CelebA dataset [23] which has been adopted by most relevant works [7, 11, 17, 26].

Dataset and preprocessing. The CelebA dataset [23] contains 202,599 aligned facial images cropped to 178×218 , with 40 *with/without* attribute labels for each image. The images are divided into training set, validation set and test set. We take 1,000 images from the validation set to assess the training process, use the rest of the validation set and the training set to train our STGAN model, and utilize the test set for performance evaluation. We consider 13 attributes, including *Bald*, *Bangs*, *Black Hair*, *Blond Hair*, *Brown Hair*, *Bushy Eyebrows*, *Eyeglasses*, *Male*, *Mouth Slightly Open*, *Mustache*, *No Beard*, *Pale Skin* and *Young*, due to that they are more distinctive in appearance and cover most attributes used by the relevant works. In our experiment, the central 170×170 region of each image is cropped and resized to 128×128 by bicubic interpolation. Training and inference time please refer to the suppl.

Qualitative results. We compare STGAN with four competing methods, *i.e.*, IcGAN [26], FaderNet [17], AttGAN [11] and StarGAN [7]. The qualitative results are

Method	<i>Bald</i>	<i>Bangs</i>	<i>Eyebrows</i>	<i>Glasses</i>	<i>Hair Color</i>	<i>Male</i>
AttGAN	12.76%	34.28%	10.64%	30.04%	11.52%	15.68%
StarGAN	11.28%	18.12%	19.40%	19.20%	32.28%	13.52%
STGAN	75.96%	47.60%	69.96%	50.76%	56.20%	70.80%

Method	<i>Mouth Open</i>	<i>Mustache</i>	<i>No Beard</i>	<i>Pale Skin</i>	<i>Young</i>	<i>Average</i>
AttGAN	20.40%	20.20%	18.92%	21.08%	15.16%	19.15%
StarGAN	23.40%	10.04%	20.36%	16.52%	27.92%	19.27%
STGAN	56.20%	69.76%	60.72%	62.40%	56.92%	61.58%

Table 3: Results of user study for ranking the models on facial attribute editing task.

Method	AttGAN	StarGAN	CycleGAN	STGAN
<i>summer</i> \rightarrow <i>winter</i>	4.7%	9.9%	24.9%	60.5%
<i>winter</i> \rightarrow <i>summer</i>	17.0%	7.9%	24.6%	50.5%

Table 4: Results of user study for ranking the models on season conversion task.

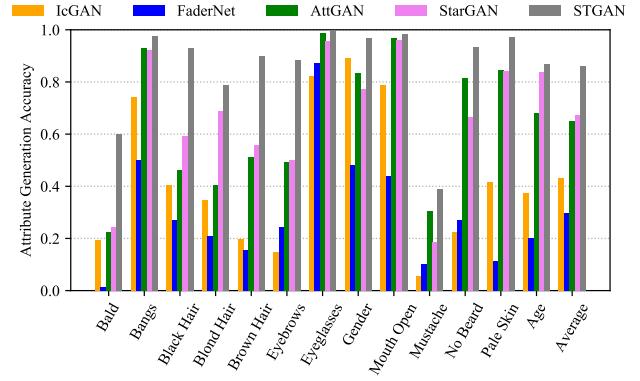


Figure 7: Attribute generation accuracy of IcGAN [26], FaderNet [17], AttGAN [11], StarGAN [7] and STGAN.

shown in Fig. 6. The results of AttGAN are generated by the released model, and we retrain other models for a fair comparison. It can be observed from Fig. 6, all the competing methods are still limited in manipulating complex attributes, *e.g.*, *Bald*, *Hair*, and *Age*, and are prone to over-smoothing results. Besides, their results are more likely to be insufficiently modified and photo non-realistic when dealing with complex and/or multiple attributes. In comparison, our STGAN is effective in correctly manipulating the desired attributes, and can produce results with high image quality. More editing results are given in the suppl.

Quantitative evaluation. The performance of attribute editing can be evaluated from two aspects, *i.e.*, image quality and attribute generation accuracy. Due to the unavailability of editing result, we resort to two alternative measures for quantitative evaluation of our STGAN. First, we use the training set of STGAN to train a deep attribute classification model which can attain an accuracy of 94.5% for the 13 attributes on the test set. Then Fig. 7 shows the attribute generation accuracy, *i.e.*, classification accuracy on the changed attributes of editing results. It can be seen that

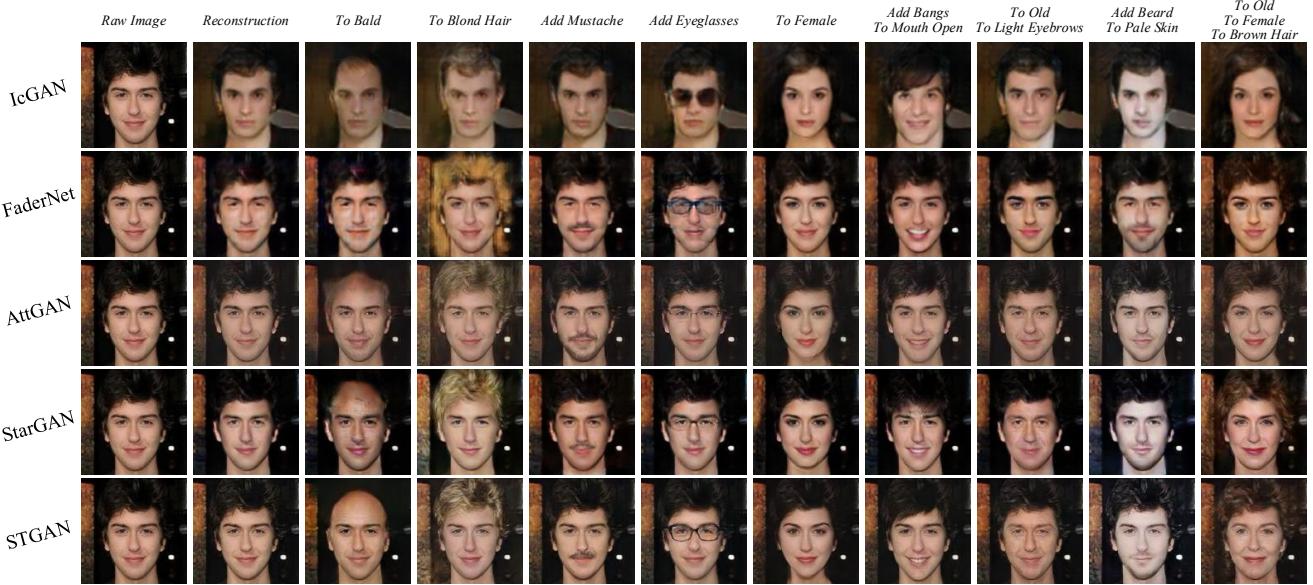


Figure 6: Facial attribute editing results on the CelebA dataset. The rows from top to down are results of IcGAN [26], FaderNet [17], AttGAN [11], StarGAN [7] and STGAN.

our STGAN outperforms all the competing methods with a large margin. For the attributes *Bald*, *Black Hair*, *Brown Hair*, and *Eyebrows*, STGAN achieves 20% accuracy gains against the competing methods.

As for image quality, we keep target attribute vector the same as the source one, and give the the PSNR/SSIM results of reconstruction in Table 2. Benefited from the STUs and difference attribute vector, our STGAN achieves much better reconstruction (> 7 dB by PSNR) in comparison to AttGAN and STGAN. The result is consistent with Fig. 2. The reconstruction ability of IcGAN is very limited due to the training procedure. FaderNet obtains better reconstruction results, mainly ascribing to that each FaderNet model is trained to deal with only one attribute.

User study. User study on a crowdsourcing platform is conducted to evaluate the generation quality of three top-performance methods, *i.e.*, AttGAN, StarGAN and STGAN. We consider 11 tasks for 13 attributes, as the transfer among *Blond Hair*, *Black Hair* and *Brown Hair* are merged into *Hair Color*. For each task, 50 validated people participate in and each of them is given 50 questions. In each question, people are given a source image randomly selected from test set and the editing results by AttGAN, StarGAN and STGAN. For a fair comparison, the results are shown in a random order. The users are instructed to choose the best result which *changes the attribute more successfully, is of higher image quality and better preserves the identity and fine details of source image*. The results are shown in Table 3, and STGAN has higher probability to be selected as the best method on all the 11 tasks.

4.2. Season Translation

We further train our STGAN for image-to-image translation between *summer* and *winter* using the dataset released by CycleGAN [34]. The dataset contains photos of Yosemite, including 1,231 summer and 962 winter images in the training set, and 309 summer and 238 winter images for testing. We also randomly select 100 images from the training set to validation. All images are used as the original size of 256×256 .

We compare our STGAN with AttGAN [11], StarGAN [7], and CycleGAN released by Zhu *et al.* [34]. Note that CycleGAN uses two generators respectively for *summer* \rightarrow *winter* and *winter* \rightarrow *summer* translation, while the other three methods conduct the two tasks with a single model. Fig. 8 shows several examples of translation results. It can be seen that STGAN performs favorably against the competing methods. We also conduct a user study using the same setting for facial attribute editing. From Table 4, our STGAN has a probability of more than 50% to win among the four competing methods.

5. Ablation Study

Using facial attribute editing, we implement several variants of STGAN, and evaluate them on CelebA [23] to assess the role of difference attribute vector and STUs. Concretely, we consider six variants, *i.e.*, (i) STGAN: original STGAN, (ii) STGAN-dst: substituting difference attribute vector with target attribute vector, (iii) STGAN-conv: instead of STU, applying a convolution operator by taking encoder feature and difference attribute vector as input to modify encoder feature, (iv) STGAN-conv-res: adopting

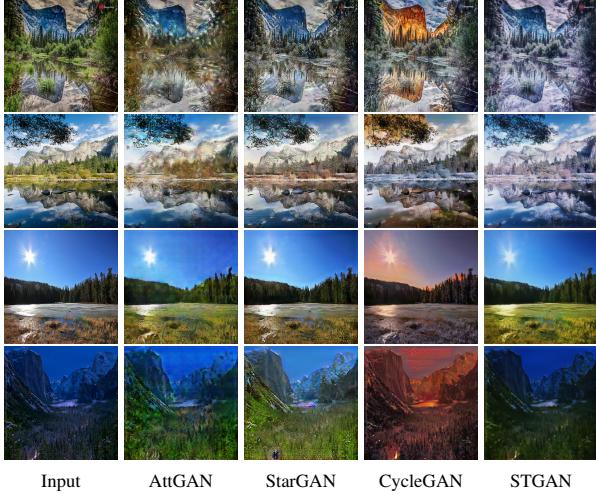


Figure 8: Results of season translation, the top two rows are *summer*→*winter*, and the bottom two rows are *winter*→*summer*.

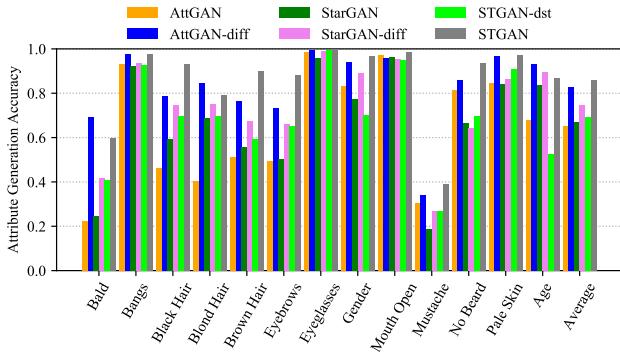


Figure 9: Effect of difference attribute vector on AttGAN, StarGAN and STGAN.

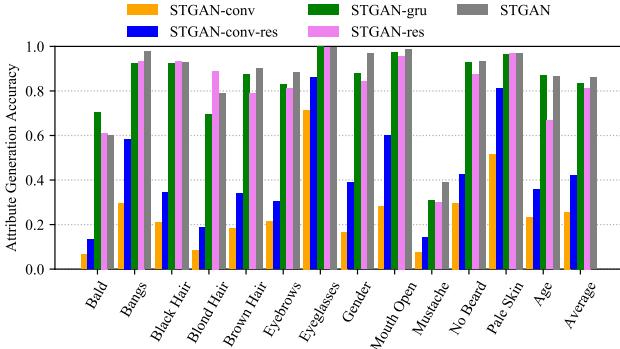


Figure 10: Attribute generation accuracy of STGAN variants.

the residual learning formulation to learn the convolution operator in STGAN-conv, (v) STGAN-gru: replacing STU with GRU in STGAN, (vi) STGAN-res: adopting the residual learning formulation to learn the STU in STGAN. We also train AttGAN and StarGAN models with difference attribute vector, denoted by AttGAN-diff and StarGAN-diff. Figs. 9 and 10 show their results on attribute manipulation. Please refer to the suppl. for qualitative results.

Difference attribute vector vs. target attribute vector. In Fig. 9, we present the comparison results of AttGAN, StarGAN and STGAN-dst with their counterparts (*i.e.*, AttGAN-diff, StarGAN-diff and STGAN) by using difference attribute vector. One can see that difference attribute vector generally benefit attribute generation accuracy for all the three models. Moreover, empirical studies show that the use of difference attribute vector gives rise to training stability as well as image reconstruction performance. Note that while AttGAN-diff and StarGAN-diff perform better than AttGAN and StarGAN, they still suffer from the poor image quality.

Selective Transfer Unit vs. its variants. Fig. 10 reports the attribute generation accuracy of several STGAN variants for transforming encoder feature conditioned on difference attribute vector. The two convolutional methods, *i.e.*, STGAN-conv and STGAN-conv-res, are significantly inferior to STGAN, indicating that they are limited in selective transfer of encoder feature. In comparison to STGAN-conv, STGAN-conv-res achieves relatively higher attribute generation accuracy. So we also compare STGAN with STGAN-res to check whether STU can be improved via residual learning. However, due to the selective ability of STUs, further deployment of residual learning cannot bring any gains for most attributes, and performs worse for several global (*e.g.*, *Gender*, *Age*) and fine (*e.g.*, *Mustache*, *Beard*) attributes. Finally, STGAN is compared with STGAN-gru by using transformed feature as hidden state. Although STGAN-gru performs better on *Bald*, STGAN is slightly superior to STGAN-gru for most attributes and the gain is notable for attributes *Gender* and *Mustache*.

6. Conclusion

In this paper, we study the problem of arbitrary image attribute editing for selective transfer perspective, and present a STGAN model by incorporating difference attribute vector and selective transfer units (STUs) in encoder-decoder network. By taking difference attribute vector rather than target attribute vector as model input, our STGAN can focus on editing the attributes to be changed, which greatly improves the image reconstruction quality and enhances the flexible translation of attributes. Furthermore, STUs are presented to adaptively select and modify encoder feature tailored to specific attribute editing task, thereby improving attribute manipulation ability and image quality simultaneously. Experiments on arbitrary facial attribute editing and season translation show that our STGAN performs favorably against state-of-the-arts in terms of attribute generation accuracy and image quality of editing results.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under grant No. 61671182 and 61872118.

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *USENIX conference on Operating Systems Design and Implementation*, pages 265–283, 2016. 6
- [2] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018. 3
- [3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2, 5
- [4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. 3
- [5] Ying-Cong Chen, Huajia Lin, Michelle Shu, Ruiyu Li, Xin Tao, Xiaoyong Shen, Yangang Ye, and Jiaya Jia. Faceletbank for fast portrait manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3541–3549, 2018. 3
- [6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014. 4
- [7] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 1, 2, 3, 4, 6, 7
- [8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 2, 3, 5
- [10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 2, 5
- [11] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Arbitrary facial attribute editing: Only change what you want. *arXiv preprint arXiv:1711.10678*, 2017. 1, 2, 3, 4, 6, 7
- [12] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems*, pages 3–10, 1994. 2
- [13] Le Hui, Xiang Li, Jiaxin Chen, Hongliang He, and Jian Yang. Unsupervised multi-domain image translation with domain-specific encoders/decoders. *arXiv preprint arXiv:1712.02050*, 2017. 3
- [14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2017. 2, 3
- [15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2
- [17] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017. 2, 3, 6, 7
- [18] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566, 2016. 2
- [19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 3
- [20] Mu Li, Wangmeng Zuo, and David Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016. 1, 3
- [21] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. In *European Conference on Computer Vision*, pages 272–289, 2018. 3
- [22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017. 3
- [23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 3, 6, 7
- [24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2
- [25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 3
- [26] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 1, 2, 3, 6, 7
- [27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 3
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing*

- and Computer-Assisted Intervention*, pages 234–241, 2015. [2](#), [3](#)
- [29] Wei Shen and Ruijie Liu. Learning residual images for face attribute manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4030–4038, 2017. [1](#), [3](#)
- [30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008. [2](#)
- [31] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. [3](#)
- [32] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *European Conference on Computer Vision*, pages 417–432, 2018. [2](#), [3](#)
- [33] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. In *European Conference on Computer Vision*, pages 150–165, 2018. [3](#)
- [34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. [1](#), [2](#), [3](#), [7](#)