StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

Yunjey Choi[1,2] Minje Choi[1,2] Munyoung Kim[2,3] Jung-Woo Ha[2] Sunghun Kim[2,4] Jaegul Choo[1,2]
[1] Korea University [2] Clova AI Research, NAVER
[3] The College of New Jersey [4] Hong Kong University of Science & Technology

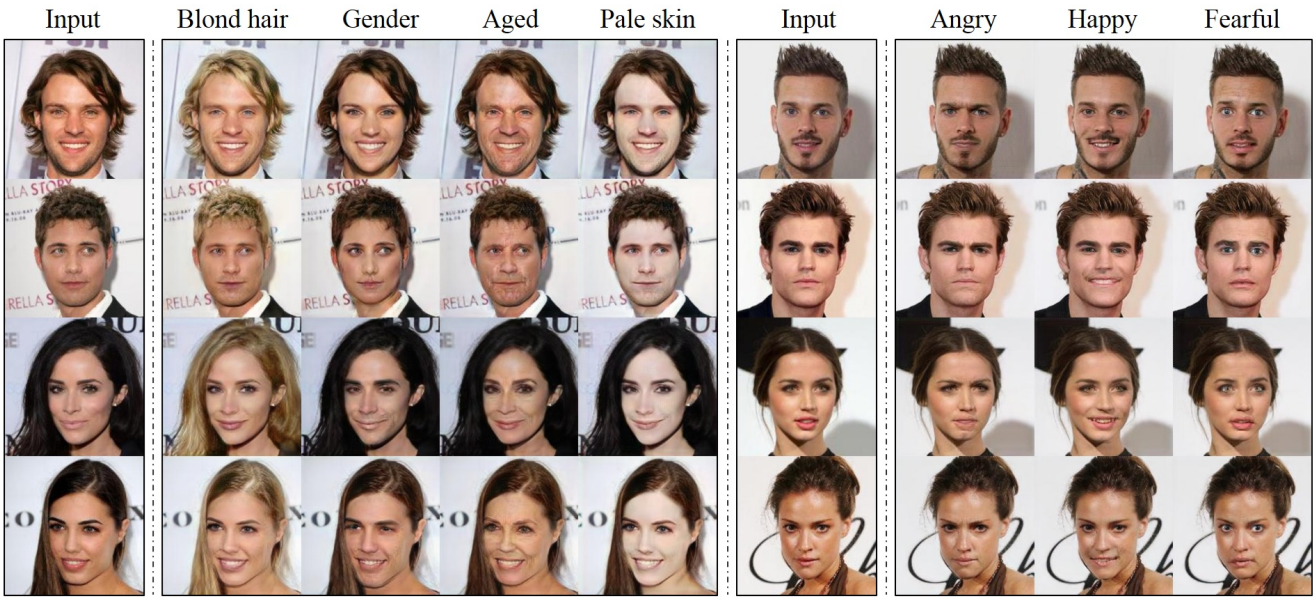

Figure 1. Multi-domain image-to-image translation results on the CelebA dataset via transferring knowledge learned from the RaFD dataset. The first and sixth columns show input images while the remaining columns are images generated by StarGAN. Note that the images are generated by a single generator network, and facial expression labels such as angry, happy, and fearful are from RaFD, not CelebA.

Abstract

Recent studies have shown remarkable success in image-to-image translation for two domains. However, existing approaches have limited scalability and robustness in handling more than two domains, since different models should be built independently for every pair of image domains. To address this limitation, we propose StarGAN, a novel and scalable approach that can perform image-to-image translations for multiple domains using only a single model. Such a unified model architecture of StarGAN allows simultaneous training of multiple datasets with different domains within a single network. This leads to StarGAN's superior quality of translated images compared to existing models as well as the novel capability of flexibly translating an input image to any desired target domain. We empirically demonstrate the effectiveness of our approach on a facial attribute transfer and a facial expression synthesis tasks.

1. Introduction

The task of image-to-image translation is to change a particular aspect of a given image to another, e.g., changing the facial expression of a person from smiling to frowning (see Fig. 1). This task has experienced significant improvements following the introduction of generative adversarial networks (GANs), with results ranging from changing hair color [8], reconstructing photos from edge maps [7], and changing the seasons of scenery images [31].

Given training data from two different domains, these models learn to translate images from one domain to the other. We denote the terms *attribute* as a meaningful feature inherent in an image such as hair color, gender or age, and *attribute value* as a particular value of an attribute, e.g., black/blond/brown for hair color or male/female for gender. We further denote *domain* as a set of images sharing the same attribute value. For example, images of women can

represent one domain while those of men represent another.

Several image datasets come with a number of labeled attributes. For instance, the CelebA[17] dataset contains 40 labels related to facial attributes such as hair color, gender, and age, and the RaFD [11] dataset has 8 labels for facial expressions such as 'happy', 'angry' and 'sad'. These settings enable us to perform more interesting tasks, namely *multi-domain image-to-image translation*, where we change images according to attributes from multiple domains. The first five columns in Fig. 1 show how a CelebA image can be translated according to any of the four domains, 'blond hair', 'gender', 'aged', and 'pale skin'. We can further extend to training multiple domains from different datasets, such as jointly training CelebA and RaFD images to change a CelebA image's facial expression using features learned by training on RaFD, as in the rightmost columns of Fig. 1.

However, existing models are both inefficient and ineffective in such multi-domain image translation tasks. Their inefficiency results from the fact that in order to learn all mappings among k domains, $k(k-1)$ generators have to be trained. Fig. 2 illustrates how twelve distinct generator networks have to be trained to translate images among four different domains. Meanwhile, they are ineffective that even though there exist global features that can be learned from images of all domains such as face shapes, each generator cannot fully utilize the entire training data and only can learn from two domains out of k. Failure to fully utilize training data is likely to limit the quality of generated images. Furthermore, they are incapable of jointly training domains from different datasets because each dataset is *partially labeled*, which we further discuss in Section 3.2

As a solution to such problems we propose StarGAN, a generative adversarial network capable of learning mappings among multiple domains. As demonstrated in Fig. 2(b), our model takes in training data of multiple domains, and learns the mappings between all available domains using only one generator. The idea is simple. Instead of learning a fixed translation (e.g., black-to-blond hair), our model takes in as inputs both image and domain information, and learns to flexibly translate the input image into the corresponding domain. We use a label (e.g., binary or one-hot vector) to represent domain information. During training, we randomly generate a target domain label and train the model to flexibly translate an input image into the target domain. By doing so, we can control the domain label and translate the image into any desired domain at testing phase.

We also introduce a simple but effective approach that enables joint training between domains of different datasets by adding a mask vector to the domain label. Our proposed method ensures that the model can *ignore* unknown labels and *focus* on the label provided by a particular dataset. In this manner, our model can perform well on tasks such as synthesizing facial expressions of CelebA images using features learned from RaFD, as shown in the rightmost columns of Fig. 1. As far as our knowledge goes, our work is the first to successfully perform multi-domain image translation across different datasets.

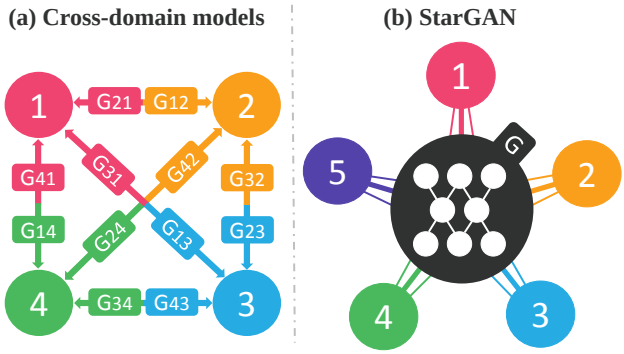

Figure 2. Comparison between cross-domain models and our proposed model, StarGAN. (a) To handle multiple domains, cross-domain models should be built for every pair of image domains. (b) StarGAN is capable of learning mappings among multiple domains using a single generator. The figure represents a star topology connecting multi-domains.

Overall, our contributions are as follows:

- We propose StarGAN, a novel generative adversarial network that learns the mappings among multiple domains using only a single generator and a discriminator, training effectively from images of all domains.
- We demonstrate how we can successfully learn multi-domain image translation between multiple datasets by utilizing a mask vector method that enables StarGAN to control all available domain labels.
- We provide both qualitative and quantitative results on facial attribute transfer and facial expression synthesis tasks using StarGAN, showing its superiority over baseline models.

2. Related Work

Generative Adversarial Networks. Generative adversarial networks (GANs) [3] have shown remarkable results in various computer vision tasks such as image generation [22, 1, 6, 30], image translation [7, 31, 8], super-resolution imaging [12], and face image synthesis [14, 24, 29]. A typical GAN model consists of two modules: a discriminator and a generator. The discriminator learns to distinguish between real and fake samples, while the generator learns to generate fake samples that are indistinguishable from real samples. Our approach also leverages the adversarial loss to make the generated images as realistic as possible.

Conditional GANs. GAN-based conditional image generation has also been actively studied. Prior studies have pro-

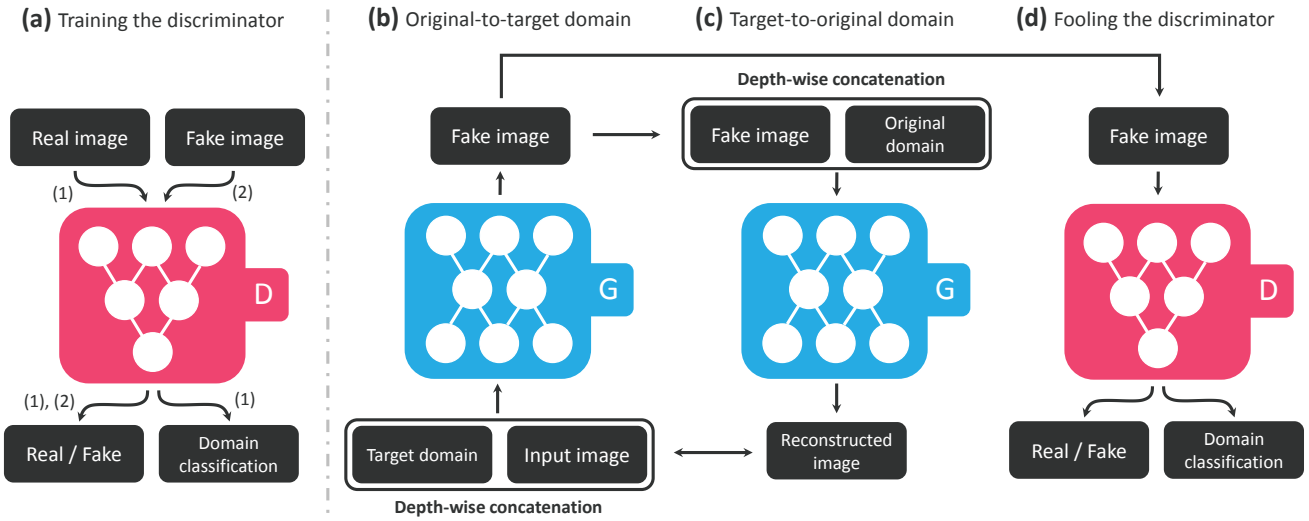

Figure 3. Overview of StarGAN, consisting of two modules, a discriminator D and a generator G. **(a)** D learns to distinguish between real and fake images and classify the real images to its corresponding domain. **(b)** G takes in as input both the image and target domain label and generates an fake image. The target domain label is spatially replicated and concatenated with the input image. **(c)** G tries to reconstruct the original image from the fake image given the original domain label. **(d)** G tries to generate images indistinguishable from real images and classifiable as target domain by D.

vided both the discriminator and generator with class information in order to generate samples conditioned on the class [19, 20, 18]. Other recent approaches focused on generating particular images highly relevant to a given text description [23, 28]. The idea of conditional image generation has also been successfully applied to domain transfer [8, 26], super-resolution imaging[12], and photo editing [2, 25]. In this paper, we propose a scalable GAN framework that can flexibly steer the image translation to various target domains, by providing conditional domain information.

Image-to-Image Translation. Recent work have achieved impressive results in image-to-image translation [7, 31, 8, 15]. For instance, pix2pix [7] learns this task in a supervised manner using cGANs[18]. It combines an adversarial loss with a L1 loss, thus requires paired data samples. To alleviate the problem of obtaining data pairs, unpaired image-to-image translation frameworks [31, 8, 15] have been proposed. UNIT [15] combines variational autoencoders (VAEs) [10] with CoGAN [16], a GAN framework where two generators share weights to learn the joint distribution of images in cross domains. CycleGAN [31] and DiscoGAN [8] preserve key attributes between the input and the translated image by utilizing a cycle consistency loss. However, all these frameworks are only capable of learning the relations between two different domains at a time. Their approaches have limited scalability in handling multiple domains since different models should be trained for each pair of domains. Unlike the aforementioned approaches, our framework can learn the relations among multiple domains using only a single model.

3. Star Generative Adversarial Networks

We first describe our proposed StarGAN, a framework to address multi-domain image-to-image translation within a single dataset. Then, we discuss how StarGAN incorporates multiple datasets containing different label sets to flexibly perform image translations using any of these labels.

3.1. Multi-Domain Image-to-Image Translation

Our goal is to train a single generator G that learns mappings among multiple domains. To achieve this, we train G to translate an input image x into an output image y conditioned on the target domain label c, $G(x, c) \rightarrow y$. We randomly generate the target domain label c so that G learns to flexibly translate the input image. We also introduce an auxiliary classifier [20] that allows a single discriminator to control multiple domains. That is, our discriminator produces probability distributions over both sources and domain labels, $D : x \rightarrow \{D_{src}(x), D_{cls}(x)\}$. Fig. 3 illustrates the training process of our proposed approach.

Adversarial Loss. To make the generated images indistinguishable from real images, we adopt an adversarial loss

$$\begin{aligned}\mathcal{L}_{adv} = &\mathbb{E}_x \left[\log D_{src}(x)\right] \; + \\ &\mathbb{E}_{x,c}[\log \left(1 - D_{src}(G(x, c))\right)],\end{aligned} \tag{1}$$

where G generates an image $G(x, c)$ conditioned on both the input image x and the target domain label c, while D tries to distinguish between real and fake images. In this paper, we refer to the term $D_{src}(x)$ as a probability distribution over sources given by D. The generator G tries to

minimize this objective, while the discriminator D tries to maximize it.

Domain Classification Loss. For a given input image x and a target domain label c, our goal is to translate x into an output image y, which is properly classified to the target domain c. To achieve this condition, we add an auxiliary classifier on top of D and impose the domain classification loss when optimizing both D and G. That is, we decompose the objective into two terms: a domain classification loss of real images used to optimize D, and a domain classification loss of fake images used to optimize G. In detail, the former is defined as

$$\mathcal{L}_{cls}^{r} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)], \tag{2}$$

where the term $D_{cls}(c'|x)$ represents a probability distribution over domain labels computed by D. By minimizing this objective, D learns to classify a real image x to its corresponding original domain c'. We assume that the input image and domain label pair (x, c') is given by the training data. On the other hand, the loss function for the domain classification of fake images is defined as

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]. \tag{3}$$

In other words, G tries to minimize this objective to generate images that can be classified as the target domain c.

Reconstruction Loss. By minimizing the adversarial and classification losses, G is trained to generate images that are realistic and classified to its correct target domain. However, minimizing the losses (Eqs. (1) and (3)) does not guarantee that translated images preserve the content of its input images while changing only the domain-related part of the inputs. To alleviate this problem, we apply a cycle consistency loss [31, 8] to the generator, defined as

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1], \tag{4}$$

where G takes in the translated image $G(x, c)$ and the original domain label c' as input and tries to reconstruct the original image x. We adopt the L1 norm as our reconstruction loss. Note that we use a single generator twice, first to translate an original image into an image in the target domain and then to reconstruct the original image from the translated image.

Full Objective. Finally, the objective functions to optimize G and D are written, respectively, as

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}_{cls}^{r}, \tag{5}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}_{cls}^{f} + \lambda_{rec}\,\mathcal{L}_{rec}, \tag{6}$$

where λ_{cls} and λ_{rec} are hyper-parameters that control the relative importance of domain classification and reconstruction losses, respectively, compared to the adversarial loss. We use $\lambda_{cls} = 1$ and $\lambda_{rec} = 10$ in all of our experiments.

3.2. Training with Multiple Datasets

An important advantage of StarGAN is that it simultaneously incorporates multiple datasets containing different types of labels, so that StarGAN can control all the labels at the test phase. An issue when learning from multiple datasets, however, is that the label information is only *partially known* to each dataset. In the case of CelebA [17] and RaFD [11], while the former contains labels for attributes such as hair color and gender, it does not have any labels for facial expressions such as 'happy' and 'angry', and vice versa for the latter. This is problematic because the complete information on the label vector c' is required when reconstructing the input image x from the translated image $G(x, c)$ (See Eq. (4)).

Mask Vector. To alleviate this problem, we introduce a mask vector m that allows StarGAN to ignore unspecified labels and focus on the explicitly known label provided by a particular dataset. In StarGAN, we use an n-dimensional one-hot vector to represent m, with n being the number of datasets. In addition, we define a unified version of the label as a vector

$$\tilde{c} = [c_1, ..., c_n, m], \tag{7}$$

where $[\cdot]$ refers to concatenation, and c_i represents a vector for the labels of the i-th dataset. The vector of the known label c_i can be represented as either a binary vector for binary attributes or a one-hot vector for categorical attributes. For the remaining $n-1$ unknown labels we simply assign zero values. In our experiments, we utilize the CelebA and RaFD datasets, where n is two.

Training Strategy. When training StarGAN with multiple datasets, we use the domain label $\tilde{c}$ defined in Eq. (7) as input to the generator. By doing so, the generator learns to *ignore* the unspecified labels, which are zero vectors, and *focus* on the explicitly given label. The structure of the generator is exactly the same as in training with a single dataset, except for the dimension of the input label $\tilde{c}$. On the other hand, we extend the auxiliary classifier of the discriminator to generate probability distributions over labels for all datasets. Then, we train the model in a multi-task learning setting, where the discriminator tries to minimize only the classification error associated to the known label. For example, when training with images in CelebA, the discriminator minimizes only classification errors for labels related to CelebA attributes, and not facial expressions related to RaFD. Under these settings, by alternating between CelebA and RaFD the discriminator learns all of the discriminative features for both datasets, and the generator learns to control all the labels in both datasets.

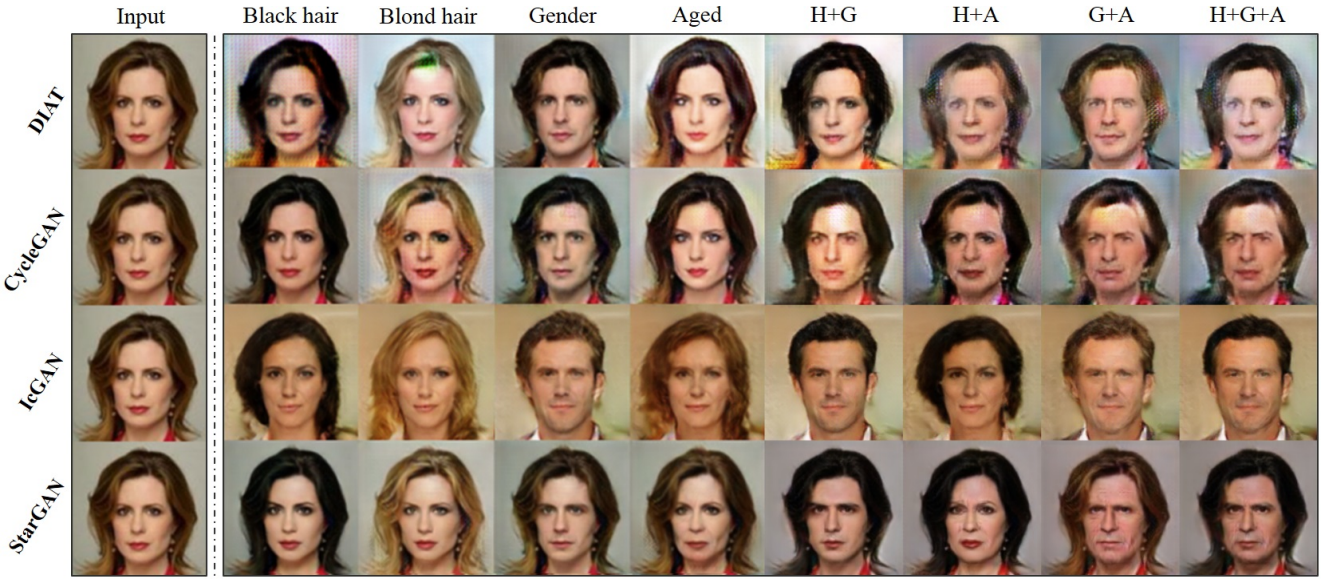

Figure 4. Facial attribute transfer results on the CelebA dataset. The first column shows the input image, next four columns show the single attribute transfer results, and rightmost columns show the multi-attribute transfer results. H: Hair color, G: Gender, A: Aged.

4. Implementation

Improved GAN Training. To stabilize the training process and generate higher quality images, we replace Eq. (1) with Wasserstein GAN objective with gradient penalty [1, 4] defined as

$$\begin{aligned}\mathcal{L}_{adv} = &\mathbb{E}_x[D_{src}(x)] - \mathbb{E}_{x,c}[D_{src}(G(x,c))] \\ &- \lambda_{gp}\,\mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}} D_{src}(\hat{x})||_2 - 1)^2],\end{aligned} \tag{8}$$

where $\hat{x}$ is sampled uniformly along a straight line between a pair of a real and a generated images. We use $\lambda_{gp} = 10$ for all experiments.

Network Architecture. Adapted from [31], StarGAN has the generator network composed of two convolution layers with the stride size of two for downsampling, six residual blocks [5], and two transposed convolution layers with the stride size of two for upsampling. We use instance normalization [27] for the generator but no normalization for the discriminator. We leverage PatchGANs [13, 7, 31] for the discriminator network, which classifies whether local image patches are real or fake. See the appendix (Section 7.2) for more details about the network architecture.

5. Experiments

In this section, we first compare StarGAN against recent methods on facial attribute transfer by conducting user studies. Next, we perform a classification experiment on facial expression synthesis. Lastly, we demonstrate empirical results that StarGAN can learn image-to-image translation from multiple datasets. All our experiments were conducted by using the model output from unseen images during the training phase.

5.1. Baseline Models

As our baseline models, we adopt DIAT [14] and CycleGAN [31], both of which performs image-to-image translation between two different domains. For comparison, we trained these models multiple times for every pair of two different domains. We also adopt IcGAN [21] as a baseline which can perform attribute transfer using a cGAN[20].

DIAT uses an adversarial loss to learn the mapping from $x \in X$ to $y \in Y$, where x and y are face images in two different domains X and Y, respectively. This method has a regularization term on the mapping as $||x - F(G(x))||_1$ to preserve identity features of the source image, where F is a feature extractor pretrained on a face recognition task.

CycleGAN also uses an adversarial loss to learn the mapping between two different domains X and Y. This method regularizes the mapping via cycle consistency losses, $||x - (G_{YX}(G_{XY}(x)))||_1$ and $||y - (G_{XY}(G_{YX}(y)))||_1$. This method requires two generators and discriminators for each pair of two different domains.

IcGAN combines an encoder with a cGAN [20] model. cGAN learns the mapping $G : \{z, c\} \rightarrow x$ that generates an image x conditioned on both the random noise z and the conditional representation c. In addition, IcGAN learns the inverse mappings $E_z : x \rightarrow z$ and $E_c : x \rightarrow c$. This allows to synthesize images conditioned on arbitrary conditional representation.

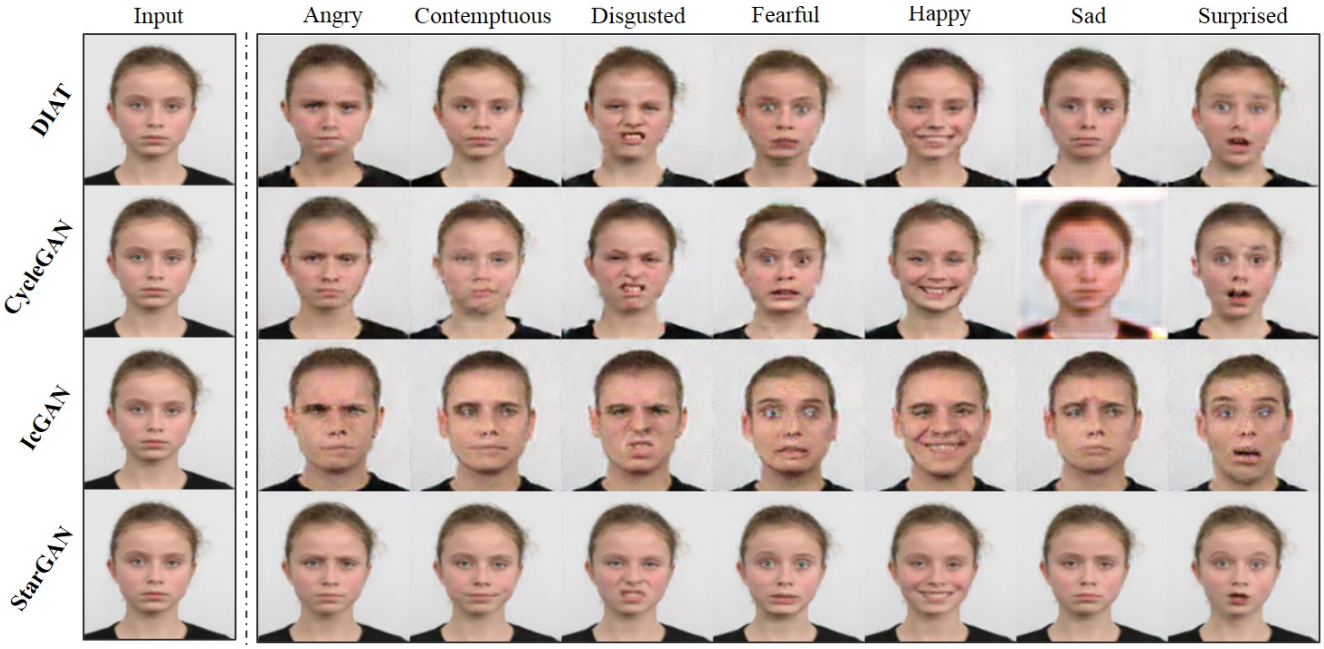

Figure 5. Facial expression synthesis results on the RaFD dataset.

5.2. Datasets

CelebA. The CelebFaces Attributes (CelebA) dataset [17] contains 202,599 face images of celebrities, each annotated with 40 binary attributes. We crop the initial 178×218 size images to 178×178, then resize them as 128×128. We randomly select 2,000 images as test set and use all remaining images for training data. We construct seven domains using the following attributes: hair color (*black*, *blond*, *brown*), gender (*male/female*), and age (*young/old*).

RaFD. The Radboud Faces Database (RaFD) [11] consists of 4,824 images collected from 67 participants. Each participant makes eight facial expressions in three different gaze directions, which are captured from three different angles. We crop the images to 256×256, where the faces are centered, and then resize them to 128×128.

5.3. Training

All models are trained using Adam [9] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. For data augmentation we flip the images horizontally with a probability of 0.5. We perform one generator update after five discriminator updates as in [4]. The batch size is set to 16 for all experiments. For experiments on CelebA, we train all models with a learning rate of 0.0001 for the first 10 epochs and linearly decay the learning rate to 0 over the next 10 epochs. To compensate for the lack of data, when training with RaFD we train all models for 100 epochs with a learning rate of 0.0001 and apply the same decaying strategy over the next 100 epochs. Training takes about one day on a single NVIDIA Tesla M40 GPU.

5.4. Experimental Results on CelebA

We first compare our proposed method to the baseline models on a single and multi-attribute transfer tasks. We train the cross-domain models such as DIAT and CycleGAN multiple times considering all possible attribute value pairs. In the case of DIAT and CycleGAN, we perform multi-step translations to synthesize multiple attributes (e.g. transferring a gender attribute after changing a hair color). All our experiments were conducted by using the model output from unseen images during the training phase.

Qualitative evaluation. Fig. 4 shows the facial attribute transfer results on CelebA. We observed that our method provides a higher visual quality of translation results on test data compared to the cross-domain models. One possible reason is the regularization effect of StarGAN through a multi-task learning framework. In other words, rather than training a model to perform a fixed translation (e.g., brown-to-blond hair), which is prone to overfitting, we train our model to flexibly translate images according to the labels of the target domain. This allows our model to learn reliable features universally applicable to multiple domains of images with different facial attribute values.

Furthermore, compared to IcGAN, our model demonstrates an advantage in preserving the facial identity feature of an input. We conjecture that this is because our method maintains the spatial information by using activation maps from the convolutional layer as latent representation, rather than just a low-dimensional latent vector as in IcGAN.

Quantitative evaluation protocol. For quantitative evalu-

Figure 6. Facial expression synthesis results of StarGAN-SNG and StarGAN-JNT on CelebA dataset.

ations, we performed two user studies in a survey format using Amazon Mechanical Turk (AMT) to assess single and multiple attribute transfer tasks. Given an input image, the Turkers were instructed to choose the best generated image based on perceptual realism, quality of transfer in attribute(s), and preservation of a figure's original identity. The options were four randomly shuffled images generated from four different methods. The generated images in one study have a single attribute transfer in either hair color (black, blond, brown), gender, or age. In another study, the generated images involve a combination of attribute transfers. Each Turker was asked 30 to 40 questions with a few simple yet logical questions for validating human effort. The number of validated Turkers in each user study is 146 and 100 in single and multiple transfer tasks, respectively.

Method	Hair color	Gender	Aged
DIAT	9.3%	31.4%	6.9%
CycleGAN	20.0%	16.6%	13.3%
IcGAN	4.5%	12.9%	9.2%
StarGAN	**66.2%**	**39.1%**	**70.6%**

Table 1. AMT perceptual evaluation for ranking different models on a single attribute transfer task. Each column sums to 100%.

Method	H+G	H+A	G+A	H+G+A
DIAT	20.4%	15.6%	18.7%	15.6%
CycleGAN	14.0%	12.0%	11.2%	11.9%
IcGAN	18.2%	10.9%	20.3%	20.3%
StarGAN	**47.4%**	**61.5%**	**49.8%**	**52.2%**

Table 2. AMT perceptual evaluation for ranking different models on a multi-attribute transfer task. H: Hair color; G: Gender; A: Aged.

Quantitative results. Tables 1 and 2 show the results of our AMT experiment on single- and multi-attribute transfer tasks, respectively. StarGAN obtained the majority of votes for best transferring attributes in all cases. In the case of gender changes in Table 1, the voting difference between our model and other models was marginal, e.g., 39.1% for StarGAN vs. 31.4% for DIAT. However, in multi-attribute changes, e.g., the 'G+A' case in Table 2, the performance difference becomes significant, e.g., 49.8% for StarGAN vs. 20.3% for IcGAN), clearly showing the advantages of StarGAN in more complicated, multi-attribute transfer tasks. This is because unlike the other methods, StarGAN can handle image translation involving multiple attribute changes by randomly generating a target domain label in the training phase.

5.5. Experimental Results on RaFD

We next train our model on RaFD to learn the task of synthesizing facial expressions. Given eight different expressions, the input domain is fixed as the 'neutral' expression, but the target domain varies among the seven remaining expressions. Therefore, the proposed task amounts to imposing a particular facial expression to a neutral facial image.

Qualitative evaluation. As seen in Fig. 5, StarGAN clearly generates the most natural-looking expressions while properly maintaining the personal identity and facial features of the input. While DIAT and CycleGAN mostly preserve the identity of the input, many of their results are shown blurry and do not maintain the degree of sharpness as seen in the input. IcGAN even fails to preserve the personal identity in the image by generating male images.

We believe that the superiority of StarGAN in the image quality is due to its implicit data augmentation effect from a multi-task learning setting. RaFD images contain a relatively small size of samples, e.g., 500 images per domain. When trained on two domains, DIAT and CycleGAN can only use 1,000 training images at a time, but StarGAN can use 4,000 images in total from all the available domains for its training. This allows StarGAN to properly learn how to maintain the quality and sharpness of the generated output.

Quantitative evaluation. For a quantitative evaluation, we compute the classification error of a facial expression on synthesized images. We trained a facial expression classifier on the RaFD dataset (90%/10% splitting for training and test sets) using a ResNet-18 architecture [5], resulting in a near-perfect accuracy of 99.55%. We then trained each of image translation models using the same training set and performed image translation on the same, unseen test set. Finally, we classified the expression of these translated images using the above-mentioned classifier. As can be seen in Table 3, our model achieves the lowest classification error, indicating that our model produces the most realistic facial expressions among all the methods compared.

Method	Classification error	# of parameters
DIAT	4.10	52.6M $\times$ 7
CycleGAN	5.99	52.6M $\times$ 14
IcGAN	8.07	67.8M $\times$ 1
StarGAN	**2.12**	**53.2M $\times$ 1**
Real images	0.45	-

Table 3. Classification errors [%] and the number of parameters on RaFD dataset.

Another important advantage of our model is the scalability in terms of the number of parameters required. The last column in Table 3 shows that the number of parameters required to learn all translations by StarGAN is seven times smaller than that of DIAT and fourteen times smaller than that of CycleGAN. This is because StarGAN requires only a single generator and discriminator pair, regardless of the number of domains, while in the case of cross-domain models such as CycleGAN, a completely different model should be trained for each source-target domain pair.

5.6. Experimental Results on CelebA+RaFD

Finally, we empirically demonstrate that our model can learn not only from multiple domains within a single dataset, but also from *multiple datasets*. We train our model jointly on CelebA and RaFD datasets using the mask vector (see Section 3.2). To distinguish between the model trained only on RaFD and the model trained on both CelebA and RaFD, we denote the former as *StarGAN-SNG* (single) and the latter as *StarGAN-JNT* (joint).

Effects of joint training. Fig. 6 shows qualitative comparisons between StarGAN-SNG and StarGAN-JNT, where the task is to synthesize facial expressions of images in CelebA. StarGAN-JNT exhibits emotional expressions with high visual quality, while StarGAN-SNG generates reasonable but blurry images with gray backgrounds. This difference is due to the fact that StarGAN-JNT learns to translate CelebA images during training but not StarGAN-SNG. In other words, StarGAN-JNT can leverage both datasets to improve shared low-level tasks such facial keypoint detection and segmentation. By utilizing both CelebA and RaFD, StarGAN-JNT can improve these low-level tasks, which is beneficial to learning facial expression synthesis.

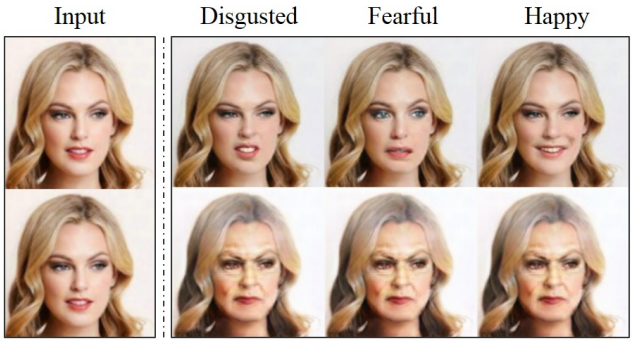

Figure 7. Learned role of the mask vector. All images are generated by StarGAN-JNT. The last row shows the result of applying the mask vector incorrectly.

Learned Role of Mask vector. In this experiment, we gave a one-hot vector input, c by setting the dimension of a particular facial expression (available from the second dataset, RaFD) to one. In this case, since the label associated with the second data set is explicitly given, the proper mask vector would be $[0, 1]$. Fig. 7 shows the case where this proper mask vector was given (the first row) and the opposite case where a wrong mask vector of $[1, 0]$ was given (the second row). When the wrong mask vector was used, StarGAN-JNT fails to synthesize facial expressions, and it manipulates the age of the input image. This is because the model ignores the facial expression label as *unknown* and treats the facial attribute label as *valid* by the mask vector. Note that since one of the facial attributes is 'young', the model translates the image from young to old when it takes in a zero vector as input. From this behavior, we can confirm that StarGAN properly learned the intended role of a mask vector in image-to-image translations when involving all the labels from multiple datasets altogether.

6. Conclusion

In this paper, we proposed StarGAN, a novel, scalable method to perform image-to-image translation among multiple domains using a single model. Besides the advantages in scalability, StarGAN generated images of higher visual quality compared to existing methods [31, 14, 21], owing to the generalization capability behind a multi-task learning setting. In addition, the proposed simple mask vector method enables StarGAN to utilize multiple datasets that are partially labeled, thus to control all available domain labels from them. In principle, our proposed model can be applied to translation between any other types of domains, e.g., style transfer, which will be one of our future work.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 214–223, 2017. 2, 5

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 3

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. 2

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 5, 6

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5, 8

[6] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 5

[8] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1857–1865, 2017. 1, 2, 3, 4

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[10] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014. 3

[11] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and Emotion*, 24(8):1377–1388, 2010. 2, 4, 6

[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[13] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, pages 702–716, 2016. 5

[14] M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016. 2, 5, 8

[15] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 3

[16] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 469–477, 2016. 3

[17] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 4, 6

[18] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[19] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. 3

[20] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585*, 2016. 3, 5

[21] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 5, 8

[22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[23] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 3

[24] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[25] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[26] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *5th International Conference on Learning Representations (ICLR)*, 2017. 3

[27] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 5

[28] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016. 3

[29] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[30] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. In *5th International Conference on Learning Representations (ICLR)*, 2017. 2

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 4, 5, 8

7. Appendix

7.1. Training with Multiple Datasets

Fig. 8 shows an overview of StarGAN when learning from both the CelebA and RaFD datasets. As can be seen at the top of the figure, the label for CelebA contains binary attributes (Black, Blond, Brown, Male, and Young), while the label for RaFD provides information on categorical attributes (Angry, Fearful, Happy, Sad, and Disgusted). The mask vector is a two-dimensional one-hot vector which indicates whether the CelebA or RaFD label is *valid*.

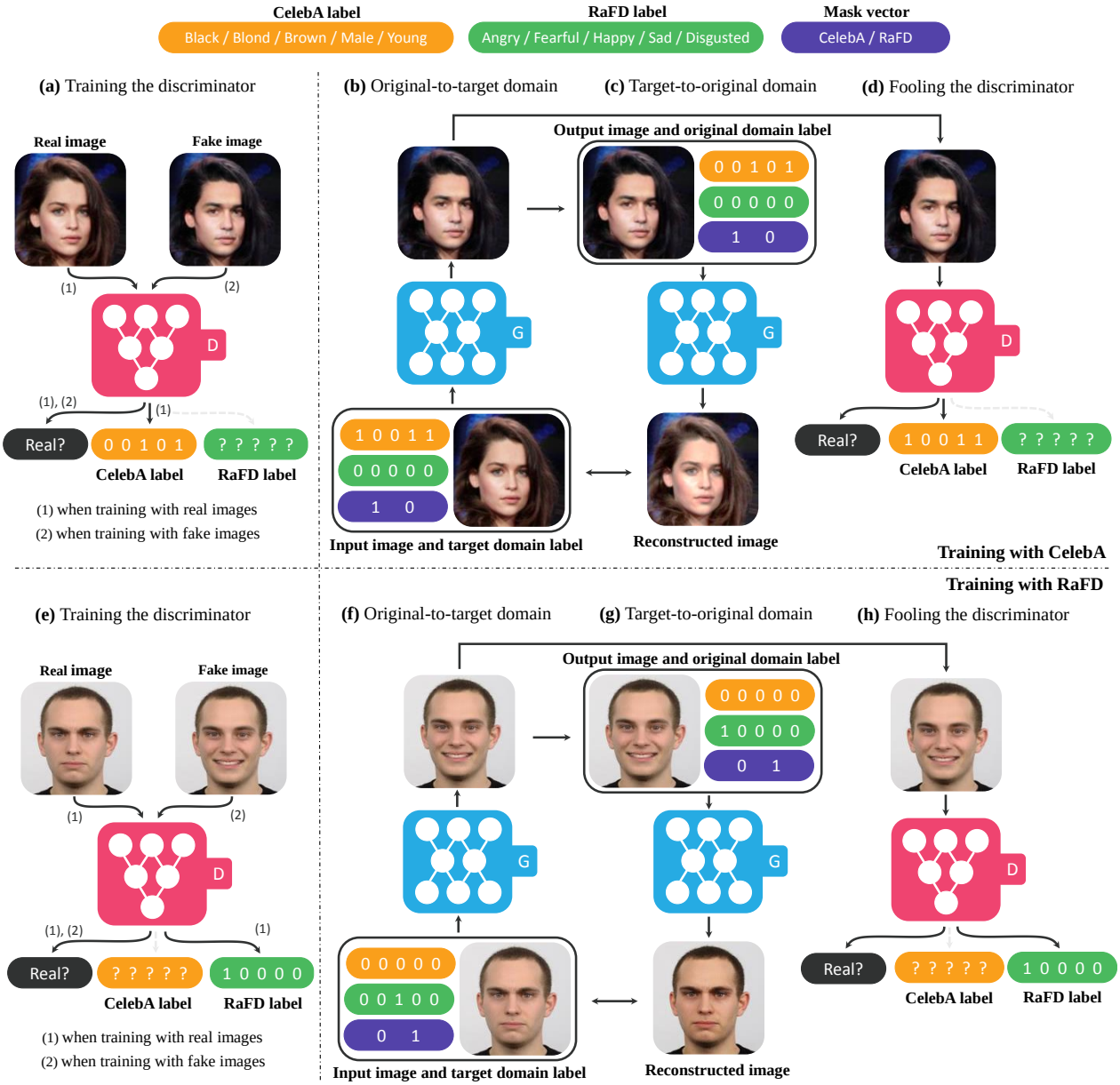

Figure 8. Overview of StarGAN when training with both CelebA and RaFD. **(a)** ∼ **(d)** shows the training process using CelebA, and **(e)** ∼ **(h)** shows the training process using RaFD. **(a), (e)** The discriminator D learns to distinguish between real and fake images and minimize the classification error only for the known label. **(b), (c), (f), (g)** When the mask vector (purple) is [1, 0], the generator G learns to focus on the CelebA label (yellow) and ignore the RaFD label (green) to perform image-to-image translation, and vice versa when the mask vector is [0, 1]. **(d), (h)** G tries to generate images that are both indistinguishable from real images and classifiable by D as belonging to the target domain.

7.2. Network Architecture

The network architectures of StarGAN are shown in Table 4 and 5. For the generator network, we use instance normalization in all layers except the last output layer. For the discriminator network, we use Leaky ReLU with a negative slope of 0.01. There are some notations; n_d: the number of domain, n_c: the dimension of domain labels ($n_d + 2$ when training with both the CelebA and RaFD datasets, otherwise same as n_d), N: the number of output channels, K: kernel size, S: stride size, P: padding size, IN: instance normalization.

Part	Input → Output Shape	Layer Information
Down-sampling	$(h, w, 3 + n_c) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), IN, ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), IN, ReLU
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	DECONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 64)$	DECONV-(N64, K4x4, S2, P1), IN, ReLU
	$(h, w, 64) \rightarrow (h, w, 3)$	CONV-(N3, K7x7, S1, P3), Tanh

Table 4. Generator network architecture

Layer	Input → Output Shape	Layer Information
Input Layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048, K4x4, S2, P1), Leaky ReLU
Output Layer (D_{src})	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3x3, S1, P1)
Output Layer (D_{cls})	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, n_d)$	CONV-(N(n_d), K$\frac{h}{64}$x$\frac{w}{64}$, S1, P0)

Table 5. Discriminator network architecture

7.3. Additional Qualitative Results

Figs. 9, 10, 11, and 12 show additional images with 256×256 resolutions generated by StarGAN. All images were generated by a single generator trained on both the CelebA and RaFD datasets. We trained StarGAN on a single NVIDIA Pascal M40 GPU for seven days.

Figure 9. Single and multiple attribute transfer on CelebA (Input, Black hair, Blond hair, Brown hair, Gender, Aged, Hair color + Gender, Hair color + Aged, Gender + Aged, Hair color + Gender + Aged).

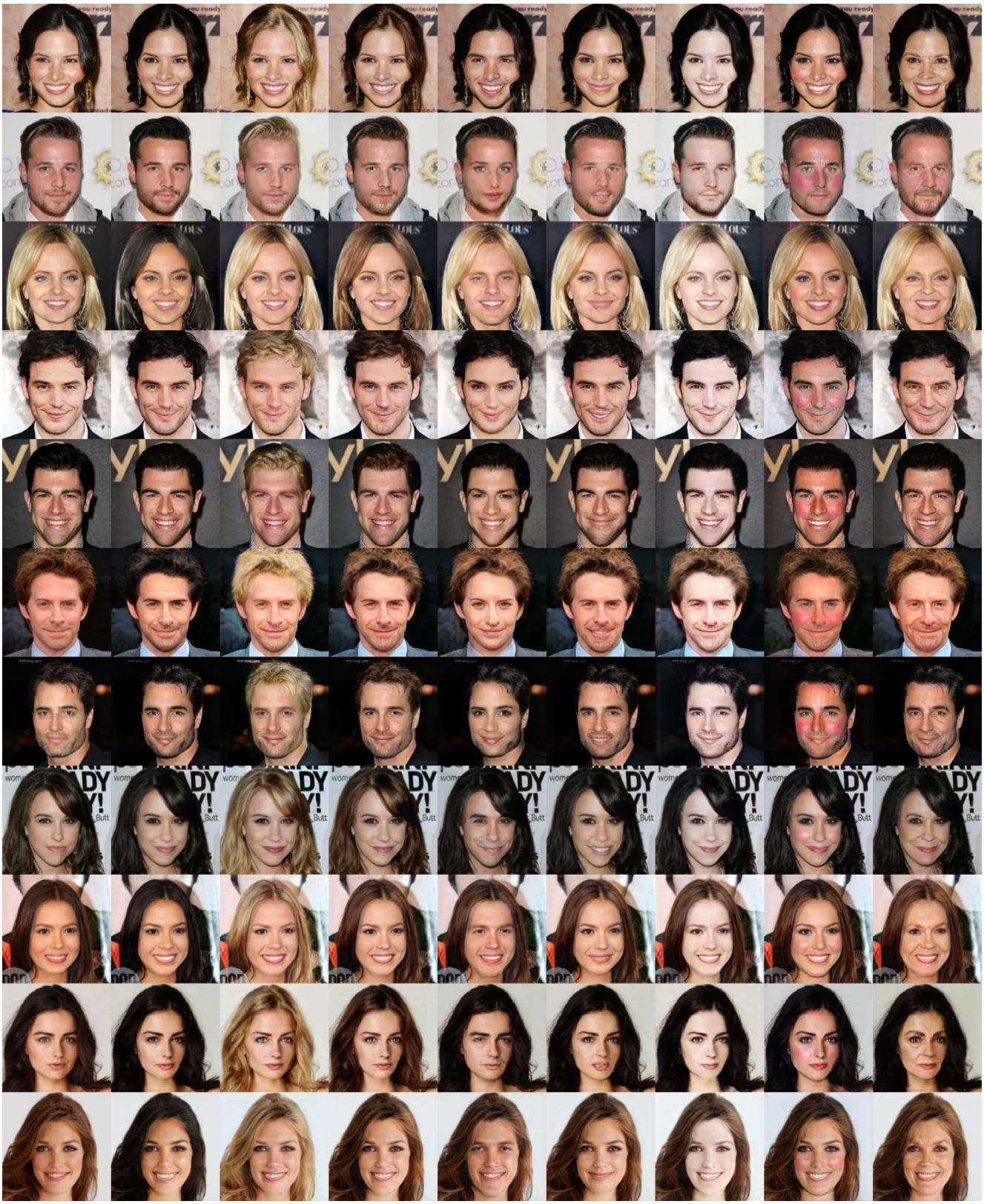

Figure 10. Single attribute transfer on CelebA (Input, Black hair, Blond hair, Brown hair, Gender, Mouth, Pale skin, Rose cheek, Aged).

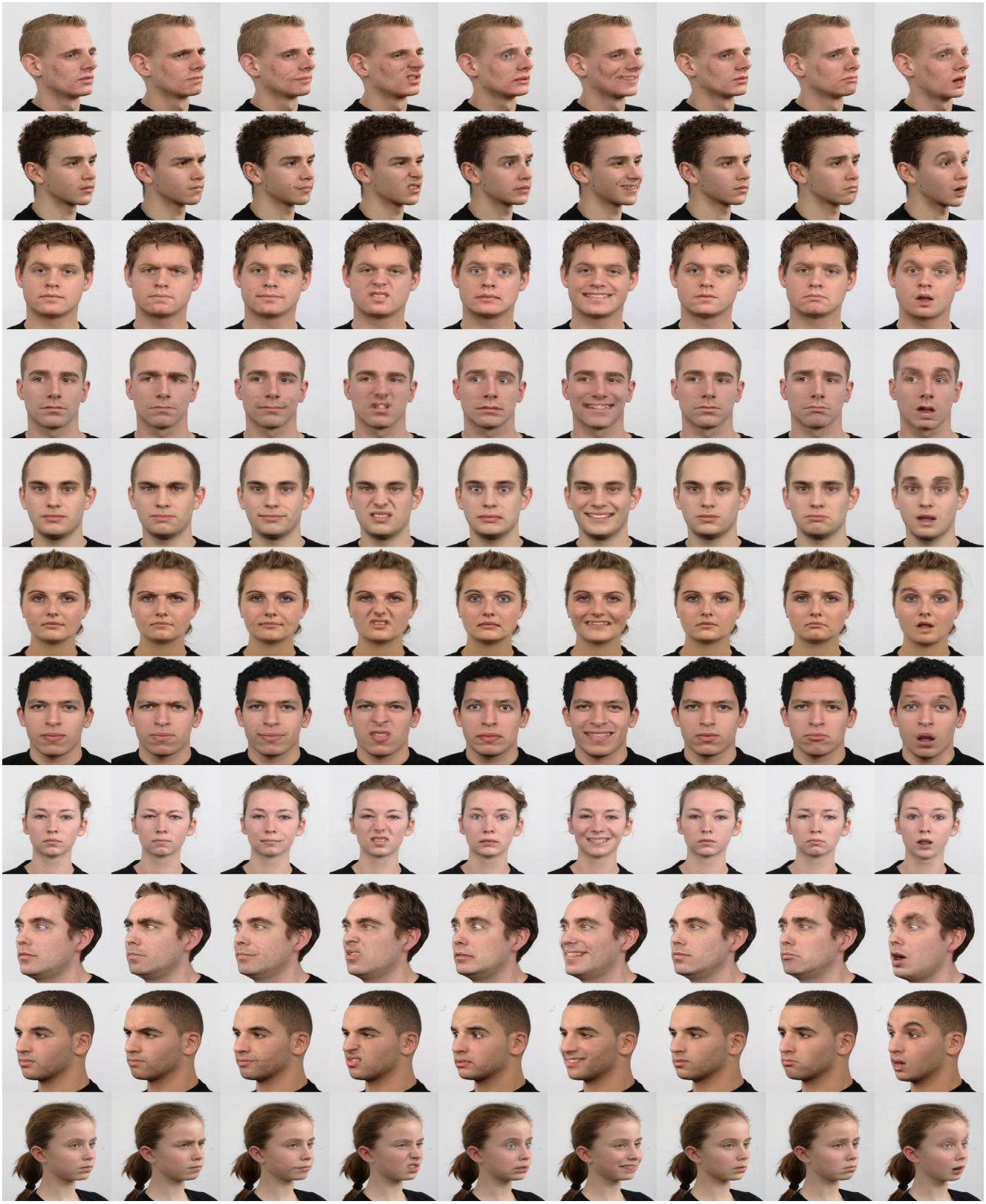

Figure 11. Emotional expression synthesis on RaFD (Input, Angry, Contemptuous, Disgusted, Fearful, Happy, Neutral, Sad, Surprised).

Figure 12. Emotional expression synthesis on CelebA (Input, Angry, Contemptuous, Disgusted, Fearful, Happy, Neutral, Sad, Surprised).