SemiStarGAN: Semi-Supervised Generative Adversarial Networks for Multi-Domain Image-to-Image Translation*

Shu-Yu Hsu[1][0000−0003−4916−9926], Chih-Yuan Yang[1][0000−0002−8989−501X], Chi-Chia Huang[2][0000−0002−8517−5525], and Jane Yung-jen Hsu[1,2][0000−0002−2408−4603]

[1] Computer Science and Information Engineering, National Taiwan University
{r05922059,yangchihyuan,yjhsu}@ntu.edu.tw
[2] Graduate Institute of Networking and Multimedia, National Taiwan University
d01944003@ntu.edu.tw

Abstract. Recent studies have shown significant advance for multi-domain image-to-image translation, and generative adversarial networks (GANs) are widely used to address this problem. However, existing methods all require a large number of domain-labeled images to train an effective image generator, but it may take time and effort to collect a large number of labeled data for real-world problems. In this paper, we propose SemiStarGAN, a semi-supervised GAN network to tackle this issue. The proposed method utilizes unlabeled images by incorporating a novel discriminator/classifier network architecture Y model, and two existing semi-supervised learning techniques—pseudo labeling and self-ensembling. Experimental results on the CelebA dataset using domains of facial attributes show that the proposed method achieves comparable performance with state-of-the-art methods using considerably less labeled training images.

Keywords: Image-to-Image Translation · Generative Adversarial Network · Semi-Supervised Learning.

1 Introduction

Image-to-image translation is the study of converting an image's representation, e.g., its colors and tone, painting style, or objects' attributes such as an identity's gender. Numerous GAN-based methods have been proposed in the literature to address this problem [6, 23, 29, 7, 15, 25, 14, 1] because of the capability provided by GAN architecture to generate realistic images to achieve effective translation.

Existing GAN-based image translation methods are designed for various purposes such as keeping consistency between source and target images [29, 7, 25]

* This research was supported in part by the Ministry of Science and Technology of Taiwan (MOST 107-2633-E-002-001, 106-2218-E-002-043, 107-2811-E-002-018), National Taiwan University (NTU-107L104039), Intel Corporation, and Delta Electronics.

or sharing same semantic space [14]. But most of them take only a source and target domain into consideration except StarGAN [1], which is the first method designed for directly translating images from a source domain to multiple target domains. The success of StarGAN relies on a prerequisite that a large number of domain-labeled training images are available. Although it is a common assumption for many existing GAN-based methods, it may require a high level of cost and effort to generate labels for real-world problems. Therefore, there will be a great merit to develop a method which requires less labeled data but achieves comparable performance.

Such a motivation has been widely adopted by many semi-supervised methods, from which we integrate two existing techniques—pseudo labeling and self-ensembling [9]—into our method addressing the multi-domain image-to-image translation problem.

Furthermore, we propose a novel type of architecture of the discriminator/classifier component in a GAN framework. We name it Y model due to its shape—a few sharing base layers plus two separated branches of layers: one for the discriminator and another for the classifier. Such a design is motivated by an observation that a discriminator and classifier serve for different purposes, so that they are likely to be better optimized if they have individual branches of layers. However, because low-level image features extracted by CNNs are highly similar such as edges, corners and spots, we retain a few common low-level layers to reduce the total number of weights and increase the stability of the training process.

We evaluate the proposed method using the CelebA dataset [16] for facial attributes synthesis and measure its performance by three metrics: Inception accuracy, human perceptual study, and classification accuracy of the auxiliary classifier. Our experimental results show that SemiStarGAN uses less labeled training images and generates favorable translated images and performance comparable with a state-of-the-art method, StarGAN.

In short, our contributions are threefold:

- We propose a novel method, which first utilizes semi-supervised learning to exploit unlabeled data for multi-domain image-to-image translation.
- We introduce a novel partially-splitting discriminator/classifier model, designed to improve the auxiliary classifier's accuracy and reduce the uncertainty caused by unlabeled images.
- We conduct experiments on hair color translation and validate the effectiveness of the proposed method, which generates higher Inception and classification accuracy by using merely one-third labeled images of a compared supervised method.

2 Related Work

Generative Adversarial Network. Generative adversarial networks (GANs) [4] are capable of generating various high-quality images so that they are widely

used to tackle assorted computer vision problems including text to image synthesis [19, 27], image-to-image translation [6, 23, 29, 1], image editing [28], and image super-resolution [10]. There are two fundamental parts contained in GANs: a generator and discriminator. During the loops of training, the former improves its ability to generate realistic images difficult to distinguish, but the latter also improves its expertise to make better judgments. As a result at the end of the training process, an effective generator is trained to work for a specific task.

GAN-based Image-to-Image Translation. Due to the effectiveness of handling complicated translation functions as a generator under a GAN framework, many GAN-based image-to-image translation methods have been proposed to generate promising results. Pix2pix [6] regulates a conditional GAN (cGAN) by a pixel-wise L1 loss to reduce the difference between a translated image and a target image. However, the method requires many highly matched pairs of images of both source and target domains to train its network, and it is a challenging preparation. To free this constraint of paired training images, two different approaches have been proposed. One of them introduces a cycle consistency loss to enable direct image translation [29, 25, 7], and the other utilizes a shared semantic space [15, 23, 14] and trains two image encoders and two decoders. Training images from both source and target domains are projected into the shared semantic space by the two encoders, and then get reconstructed by the two decoders. As the foregoing methods deal with bi-domain translation, StartGAN is the first method designed for multi-domain translation. StartGAN uses an auxiliary classifier to guide its generator so that it can sidestep the trouble of using multiple bi-domain translation. However, all of these GAN-based methods require a large amount of domain-labeled data, which take cost and effort to collect. Unlike those approaches mentioned above, our framework can reduce the need of labeled data and achieve comparable performance.

We would like to point out the detailed difference of the terms *unsupervised* used by two existing methods [14, 29] and *semi-supervised* used by the proposed method. The authors call their method unsupervised because they no longer need to use paired training images, which means that their training data are unsupervised on the pixel level. In contrast, the proposed method utilizes both labeled and unlabeled images so that its learning process is semi-supervised on the domain level.

Semi-Supervised Learning. Compared with fully supervised learning methods, semi-supervised learning (SSL) methods take advantage of unlabeled data and show promising results in many real-world problems because it is more feasible to collect a large number of unlabeled data than labeled ones. There are several regularization-based and GAN-based SSL methods in the literature highly related to the proposed method. The two methods temporal ensembling and self-ensembling [9] are proposed to develop a robust classifier which can work reliably against various perturbations. To validate this idea, their authors introduce stochastic augmentations to distort their input data, and dropouts

to enhance their classifier so that the classifier can make consistent predictions under highly altered source data. Similarly, virtual adversarial training [17] regularizes a classifier with images added with virtual adversarial perturbations.

In order to cope with unlabeled data, many GAN-based methods have been proposed by changing their objectives of output images or adversarial losses. CatGAN [21] extends a binary discriminator to a multi-class classifier and replaces a conventional GAN objective with entropy minimization of unlabeled data. SGAN [18], ImprovedGAN [20] and BadGAN [2] add an auxiliary classifier aside the discriminator. Although the classifier of these GAN- based methods can achieve favorable classification accuracy, their generated images are blurred and incomprehensible. To address this issue, TripleGAN [12] proposes two independent parts—a discriminator and classifier—to prevent their conflict of learning optimized CNN weights, which also prevents the generator from generating incomprehensible output images. In short, we adopt the techniques used by self-ensembling into the proposed method, and inspired by SGAN and TripleGAN, we design a novel Y model of the discriminator/classifier architecture, which owns a few shared layers of the discriminator and auxiliary classifier but two separated branches to reach a balance between classification accuracy and training stability.

3 Proposed Method

The proposed method, SemiStarGAN, is motivated by an existing method StarGAN [1] and address the issue of taking advantage of unlabeled training images. The success of StarGAN relies on a prerequisite that abundant domain-labeled training images are available to train its classifier to generate correct labels to give its generator the information whether the generated images contain expected attributes. However, it may take time and effort to collect a large number of labeled data for real-world tasks, and two problems will come up if only limited labeled data are available. First, its generator may create unexpected output images such as broken or over-blurred due to incorrect data models learned from insufficient training examples. Second, there may be an over-fitting problem of the auxiliary classifier because of the limited number of training examples. As a result, the generator will wrongly translate an image to an unexpected target domain. To address this problem, we propose a method which utilizes unlabeled images and considerably less labeled training images to achieve comparable performance. We will first introduce the notations widely used in GAN-based image-to-image translation methods, and then explain the details of each component contained in the proposed SemiStarGAN.

3.1 Formulation

We define the problem of multi-domain image-to-image translation as the following. Let X be a partially labeled image set, C be the label set of X on multiple domains, and X_L be the subset of X in which every image is well labeled, x

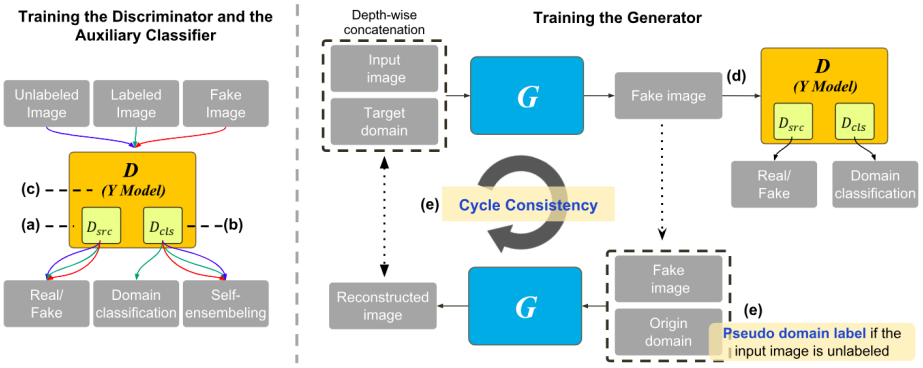

Fig. 1: Training of SemiStarGAN. (a) The discriminator D_{src} learns to distinguish between real and fake images. (b) The auxiliary classifier D_{cls} learns to correctly classify a labeled image by generating a right domain label and we enhance the classifier's robustness by applying self-ensembling which uses unlabeled data. (c) We propose Y model, a novel parameter-sharing structure between D_{src} and D_{cls} for stabilizing the training process. (d) Given an image and a target domain label, the generator G learns to generate a fake image that can fool D_{src} but still be correctly classified by D_{cls}. (e) G remains the cycle consistency by translating a fake image back to the image's original domain and a pseudo domain label is used if the input image is unlabeled.

be an image in X, and c be the label of x if available. Given the training set X and C, the problem is about developing an image translation function which generates a new image y from a given x in the training set X with a parameter of a target domain label c' and the new image y will be classified as being of the domain label c'. Multi-domain GAN-based image-to-image translation methods contains three major components: an image generator G to generate translated images, an image discriminator D_{src} to tell the difference between real/fake images, and an auxiliary classifier D_{cls} to indicate domain labels. To use both labeled and unlabeled images to train an effective image generator for multi-domian image translation, we propose a training process as illustrated in Fig. 1 and its architecture is shown in the supplementary material.

3.2 GAN Objectice

To make generated images realistic, we adopt the GAN objective

$$\mathcal{L}_{GAN} = \mathbb{E}_x[D_{src}(x)] - \mathbb{E}_{x,c'}[D_{src}(G(x,c'))] - \lambda_{gp}\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D_{src}(\hat{x})\| - 1)^2], \quad (1)$$

where c' is a given target domain label to generate a fake image $G(x,c')$, λ_{gp} is a weighting parameter to balance a gradient penalty term $\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D_{src}(\hat{x})\| - 1)^2]$ which increases the stability of a GAN's training process [5]. The symbol $\hat{x}$ stands for any image linearly mixed by x and $G(x,c')$ in their image space. The generator G aims to minimize this objective while the discriminator D_{src} tries to maximize it.

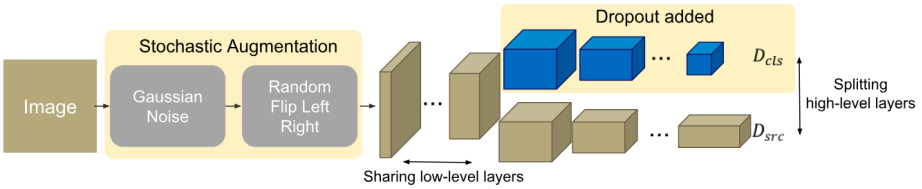

Fig. 2: The proposed Y model architecture and self-ensembling components of the classifier. The discriminator D_{src} and the auxiliary classifier D_{cls} share a few early-stage layers for extracting common low-level features and then split into two branches for learning their individual high-level features. For the two stochastic augmentations required by self-ensembling, we propose to use the Gaussian noise and random horizontal flipping. We only apply the dropouts used by self-ensembling to D_{cls}.

3.3 Domain Classification Loss and Self-Ensembling

The auxiliary classifier D_{cls} is designed to let G know whether generated images $G(x, c')$ own expected attributes so that they can be correctly recognized. That is, D_{cls} is utilized to optimize G. Hence, it is essential to improve the classification accuracy of D_{cls} and we design a labeled images' classification loss

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

which penalizes D_{cls} if it wrongly predicts x's domain label.

But for a training set containing many unlabeled images, we propose to integrate established learning techniques to improve the classifiers robustness. To do it, we adopt self-ensembling [9], a method generates robust prediction results by using two stochastic augmentations to alter its input stimulus and adding a few dropouts to enhance its classifier. The choices of the two stochastic augmentations are made upon the input data, and we select Gaussian noise and random horizontal flipping for our experiments conducted on facial attribute domains, which remain consistent after either augmentation. Fig. 2 shows the proposed auxiliary classifier and its self-ensembling components, and we define a domain classification loss for unlabeled images

$$\begin{aligned}\mathcal{L}_{cls}^{u} = &\mathbb{E}_{x}[\|D_{cls}(\phi(x,\epsilon)) - D_{cls}(\phi(x,\epsilon'))\|_2^2 \\ &+ \mathbb{E}_{x,c'}[\|D_{cls}(\phi(G(x,c'),\epsilon)) - D_{cls}(\phi(G(x,c'),\epsilon'))\|_2^2],\end{aligned} \tag{3}$$

where ϕ is the stochastic augmentation function, and ϵ and ϵ' are two different parameter settings of ϕ for generating different augmentation. Self-ensembling not only is applied to the unlabeled images but also to the fake images $G(x, c')$ and labeled images. To make G translate images to target domains correctly, we adopt domain classification loss for fake images

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|G(x,c'))], \tag{4}$$

which punishes G if its fake image $G(x, c')$ is not classified as the domain c'.

3.4 Cycle Consistency and Pseudo Cycle Consistency Loss

We aim to develop a network which can not only generate realistic images containing correct domain attributes, but also prevent random permutation often occurring in large networks. Thus we adopt an approach of remaining cycle consistency [29] and define a consistency loss

$$\mathcal{L}_{rec}^{l} = \mathbb{E}_{x,c,c'}[\|x - G(G(x, c'), c)\|_1], \tag{5}$$

which regulates the generator G by translating a generated image $G(x, c')$ back to its origin domain c and encouraging the new image $G(G(x, c'), c)$ to be similar with the original image x. To apply cycle consistency for unlabeled images which do not have domain labels, we define a loss of pseudo cycle consistency

$$\mathcal{L}_{rec}^{u} = \mathbb{E}_{x,c'}[\|x - G(G(x, c'), D_{cls}(x))\|_1], \tag{6}$$

where we utilize the auxiliary classifier D_{cls} to predict unlabeled data's labels, which are required to brought into a loss of cycle consistency. Our use of pseudo labels is inspired by an existing method, pseudo labeling [11], but we do not follow its approach of joining genuine and pseudo labeled images to train a classifier. We only use pseudo labels to enable cycle consistency for unlabeled data.

3.5 Y Model: Splitting Classifier and Discriminator.

Many existing methods design their classifiers and discriminators in a convenient way that both classifier and discriminator share most neural network layers except the last one. Such a type of straightforward architecture is based on a hypothesis that a common set of image features from low- to high-level is sufficient for both classifier and discriminator, but it remains an open question under which situation the hypothesis works.

Based on an observation that a discriminator and auxiliary classifier serve for different purposes, i.e. telling real/fake images and predicting domain labels, and another observation that adopting self-ensembling must create a new sort of network architecture, we propose a partially splitting model of the classifier and discriminator, named Y model due to its shape similar to the letter Y, as shown in Fig. 2.

The architecture of Y model is inspired from TripleGAN, which totally splits its discriminator and classifier. However, since most convolutional neural networks extract similar low-level image features such as edges, corners, and spots [26], we propose to share a few common layers in the early stage used by both discriminator and auxiliary classifier. Beyond those common layers, either the discriminator or auxiliary classifier owns its individual convolutional layers for learning specific high-level features. To the best of our knowledge, no similar architecture has been proposed in the literature to address the problem of image translation.

3.6 Full Objective

Finally, we make the overall objective for the generator G as

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}^l_{rec} + \lambda_{rec}\mathcal{L}^u_{rec} \tag{7}$$

and for the discriminator D_{scr} and the auxiliary classifier D_{cls} as

$$\mathcal{L}_D = -\mathcal{L}_{GAN} + \lambda_{cls}\mathcal{L}^r_{cls} + \lambda^u_{cls}\mathcal{L}^u_{cls} \tag{8}$$

where λ_{cls}, λ_{rec}, and λ^u_{cls} are weight parameters make losses balanced.

4 Experimental Validation

We conduct two experiments to validate the proposed method, and report its performance by two sets of numerical evaluation and one human perceptual study. The code is publicly available at GitHub[3].

CelebA_HAG image set. The CelebA dataset [16] is a widely used face image dataset, which contains 202,599 face images collected from 10,177 identities with large pose variations and background clutter. Each image in the CelebA dataset is labeled with 40 binary attributes, among which three attributes—*black hair*, *blond hair* and *brown hair* indicate hair colors, but a few images are annotated as positive in more than one of the three attributes because the hair colors of those images are between the three given colors. To create an unambiguous image set in which hair colors become mutually exclusive, we extract all images from the CelebA dataset labeled as positive for only one of the three hair color attributes, i.e., images with multiple positive hair color attributes are excluded. We name the image set CelebA_HAG, which contains 115309 images. The letters H, A, and G stand for hair color, age, and gender because we take two more attributes—young and male—into account, and the statistics of the CelebA_HAG image set is shown in Table 1. Those attributes, three hair colors, two age states (young, not young), and two genders (male, not male) are all mutually exclusive and make up 12 domains. For a fair comparison with a state-of-the-art method StarGAN, we generate images for experiments using its release code, which crops a central region of 178×178 pixels from the original CelebA images of 178×218 pixels, and then downsamples cropped images using bilinear interpolation into a smaller size of 128×128 pixels.

4.1 Evaluation Metrics

We evaluate the performance of the proposed method using three metrics: Inception accuracy, classification accuracy, and human perceptual rating.

[3] https://github.com/KevinYuimin/SemiStarGAN

Table 1: Statistics of the CelebA_HAG image sets used for the experimental validation of the proposed method.

	male +		male -		
hair color	young +	young -	young +	young -	total
black	19722	4939	21104	1568	47333
blond	1026	565	22691	4478	28760
brown	9475	2664	24091	2986	39216
total	30223	8168	67886	9032	115309

Inception Accuracy. In order to fairly access the effectiveness of a generator by its translated images, we utilize a strong classifier to objectively evaluate the saliency of the translated images [20]. We choose Inception-v3 [22] as the classifier due to its state-of-the-art performance for object recognition. We use its publicly released model pretrained on the ImageNet dataset [3], and refine it using our CelebA_HAG dataset. Each test image is translated to another domain and classified by the strong classifier. The inception accuracy Acc_{incept} is defined as the fraction of translated test images which are correctly classified by the refined Inception-v3 network. Note that though a high Acc_{incept} value means good translation saliency, it reveals little information about the quality of the translated images, so that we need human perceptual studies to measure their visual quality.

Human Perceptual Study. We carry out our human perceptual studies through Amazon Mechanical Turk (AMT). We present each translated image to 5 different turkers and ask two multiple-choice questions. First, which domain does the subject on the translated image belongs to? That is, turkers work in the same manner as Inception-v3 to classify translated images. Second, how is the quality of the translated image compared with its original image? and we offer three choices: similar, slightly worse, and worse. Because the image quality of the CelebA datset varies significantly, when we ask the second question, we present not only a translated image but also its original one. To increase the reliability of this perceptual study, we set up a criterion to hire turkers whose approve rate and submission history exceed 90% and 30 times, and accept the result of a HIT (human intelligence task) only if at least 3 of 5 turkers select the same choice for both questions.

Classification Accuracy of the Auxiliary Classifier In the proposed method, the auxiliary classifier takes an important role to guide the generator to translate images with eligible saliency. To evaluate the effectiveness of the proposed auxiliary classifier, we compute the classifier's accuracy Acc_{aux}, which is defined as the fraction of the number of correctly classified test images divided by the total number.

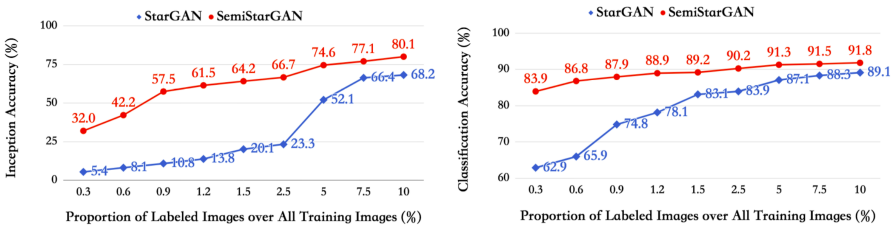

Fig. 4: Learning curves of StarGAN and SemiStarGAN generated from the experiment of the three hair color domains in terms of the Inception and classification accuracy shown in the left and right subfigures respectively. SemiStarGAN successfully exploits unlabeled images to generate translated images with higher accuracy over StarGAN for a wide range of numbers of labeled training images.

4.2 Implementation and Training

Implementation Our generator use the same CNN architecture as [1], which is composed of three convolutional downsampling layers, six residual blocks and three transposed convolutional layers for upsampling. Each convolutional layer except the last one is followed by a step of instance normalization [24]. Our discriminator and auxiliary classifier share the first two convolution layers and then split into two branches. Our discriminator branch, similar to StarGAN's discriminator, contains four convolutional layers and PatchGANs [6, 29, 1, 13], which divides an image into several patches and discriminates between real and fake patches. Our auxiliary classifier branch contains six convolutional layers, three dropout layers, and one 1×1 convolutional layer. More details of our network architecture is shown in the supplementary material.

Training Detail We initialize SemiStarGAN using random numbers and train it using batches of eight labeled samples and eight unlabeled samples in 250K iterations (20 epochs including unlabeled data). We use the Adam optimizer [8] to train SemiStarGAN and set the optimizer's two exponential decay rates for the moment estimates β_1 and β_2 as 0.5 and 0.999 respectively.

We set the initial learning rate as 0.0001 and keep it unchanged for the first half of iterations, and linearly reduce it to 0 for the second half. Regarding the parameters of the proposed method, we set the classification weight λ_{cls} as 1, gradient penalty weight λ_{gp} as 10, and reconstruction weight λ_{rec} as 10. We use Gaussian ramp up weighting function proposed by self-ensembling to set the classification weight for unlabeled images λ^u_{cls} as 2 for the first one-third training iterations, and reduce it to 0 for the last one-third training images.

4.3 Experimental Results

Experiment on three domains of hair colors. In this experimental setting, we only create three domains using the hair color attributes of our CelebA_HAG

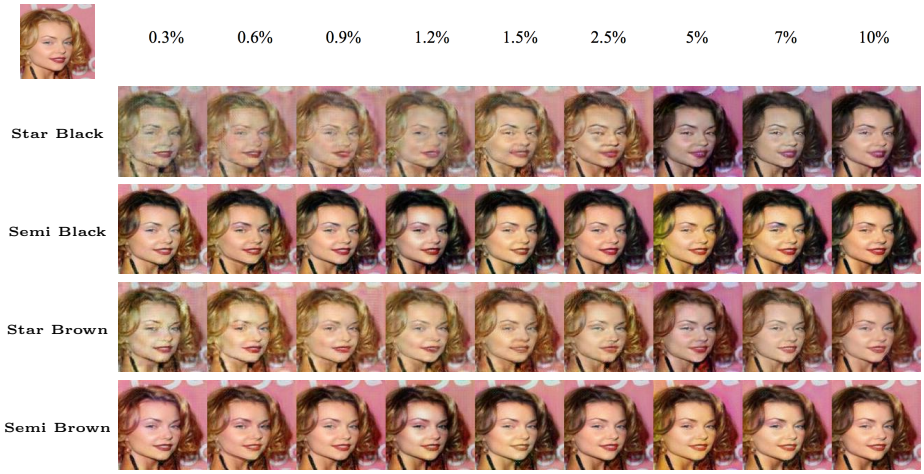

Fig. 5: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over three hair color domains. (Top-left) A source image belonging to the domain of blond hair. (Top) Proportions of labeled images over all training images of the unit of percentage. (Left) The method and the target domain. StarGAN needs 5% labeled images while SemiStarGAN only needs 1.2% to achieve desirable translation saliency and quality.

image set in order to assure large numbers of samples available in all domains to investigate the performance of the proposed method affected by the numbers of labeled training images. We randomly split the CelebA_HAG image set into training and test sets where the test set contains 5001 images (1667 images per domain), and the training set contains the remaining 110308 images. We use all images in the training set with their hair color labels to train the Inception-v3-based strong classifier and reach an accuracy rate of 92.4%. We randomly select 11031 images (3677 images per domain, 10% of all training images) to make up a labeled training set, and use the remaining 99277 (90%) images as the unlabeled training set. From the maximal proportion as 10%, we gradually reduce the number but keep the three domains balanced to do a series of experiments. In order to reduce the uncertainty caused by random selection, we repeat the experiments 3 times using 3 different seed numbers (0,1,2) and report their averaged Inception and classification accuracy in Fig. 4, and a set of translated images in Fig. 5. For a fair comparison with a state-of-the-art method StarGAN, we use its publicly released code and original setting to train its network on our CelebA_HAG dataset. Using the same labeled training images, the proposed SemiStarGAN generates higher accuracy rates over StarGAN because the SemiStarGAN takes advantage of a large number of unlabeled training images. Even only a small portion of labeled training images are used (0.3% of the total), due to the capability of exploiting a large portion of unlabeled images (90% of the total), the proposed method generated images with better quality

Table 2: Results of the human perceptual study of the experiment using three domains of hair colors. The #HITs indicates the number of HITs in which at least 3 of 5 turkers reach a consensus. When using only 1/3 amount (2.5% v.s. 7.5%) of labeled training images used by StarGAN, the proposed SemiStarGAN method generates higher accuracy and better image quality evaluated by AMT turkers. Note a smaller percentage of the two worse options means better quality.

		Question 1		Question 2			
Method	Prop.	Accuracy.	#HITs	Similar	Slightly Worse	Worse	#HITs
SemiStarGAN	2.5%	**64.29%**	196	**66.22%**	**21.62%**	**12.16%**	148
StarGAN	7.5%	59.90%	197	57.82%	24.49%	17.69%	147

than StarGAN. The results of our human perception study as shown in Table 2 also indicate that the proposed method generate better image quality.

Experiment on 12 domains of hair colors, age, and gender. We take gender and age into account to evaluate the proposed method under a large domain number 12. We randomly select 2400 images, 200 per domain, to make up a test image set. In order to generate multi-domain classification labels composed by 3 fields (hair color, gender, and age), we replace the two softmax layers used in Inception-v3 (one at the end of the main branch, and another at the end of its auxiliary branch, more details are reported in the supplementary material) with two sigmoid layers. We use the remaining 112909 images to train the strong classifier and reach a classification accuracy rate of 87.7%. Since the numbers of images belonging to the 12 domains are uneven as shown in Table 1, we randomly select 3600 images, 300 per domain, from the training image set as the labeled training image set, and treat the remaining 109509 images as the unlabeled images. We repeat the experiments 3 times using 3 different seed numbers (0,1,2) and show their averaged performance in Fig. 8 and qualitative comparisons in Fig. 6. The proposed method generates both higher Inception and classification accuracy rates then StarGAN.

The Effectiveness of the Y Model. To investigate the effectiveness of the Y model, we take two other models into consideration: the mixed model similar to the architecture used in StarGAN, and a model whose discriminator and classifier branches are totally separated. We name the former D/C model due to its combined architecture and the latter II model due to its shape of two independent branches. For a fair comparison, the three types of architecture use the same stochastic augmentations and dropouts, and their classification accuracy of the three hair color domains test images is shown in Fig. 9 for every interval of 500 training iterations using 0.3% labeled training data. All of the three models reach their accuracy plateaus before 6000 iterations, and the proposed Y model not only generates higher accuracy but also performs more stably than the D/C model, which shows that the partially split structure can

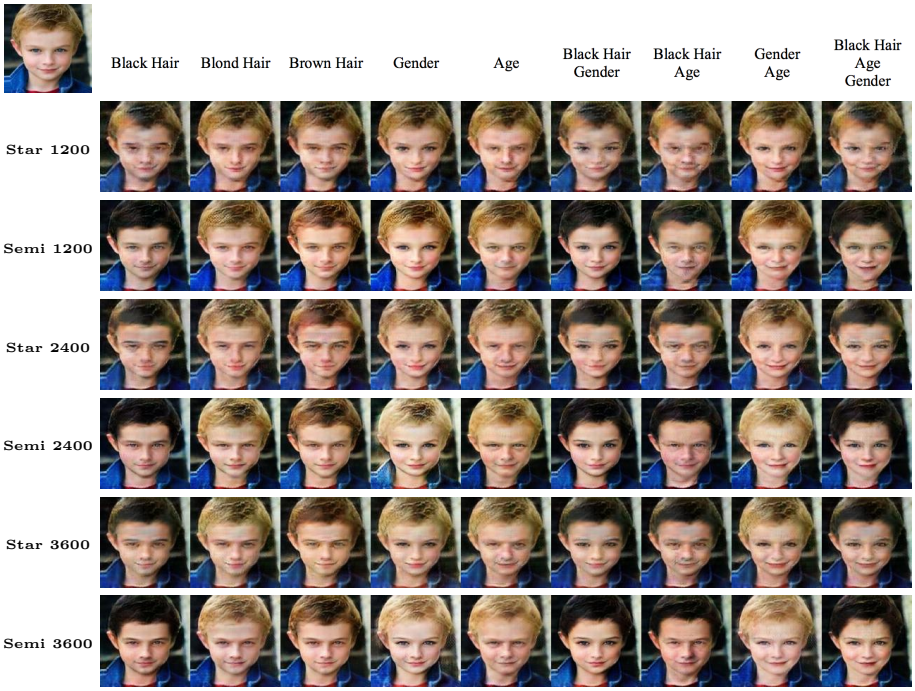

Fig. 6: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over the 12-domain problem. (Top-left) A source image belonging to the domain of blond young male. (Top) The changed facial attributes. (Left) Methods and the numbers of used labeled training images. Compared with StarGAN, SemiStarGAN generates images with better saliency in terms of expected attributes of the target domains.For example of the target domain of black hair using the same 1200 labeled training images (the two images at the first column and first two rows), SemiStarGAN generates obvious black hair, but StarGAN does not. For another example of the target domain of black hair, and different gender and age using the same 3600 labeled training images (the two image at the last column and last two rows), SemiStarGAN generates blacker hair, clearer eyes and smoother skin than StarGAN.

learn better high-level features and achieve better performance. We ascribe the superiority of the Y model over the Π model to our used GAN objective. As mentioned in several papers presenting semi-supervised GAN methods [18, 20, 2], generated samples and an adversarial loss provided by the discriminator can regularize the classifier and improve the robustness. To sum up, the Y model lets the discriminator and classifier not only learn their own suitable high-level features in the same manner of TripleGAN but also gain benefits from GAN objectives proved effective by other semi-supervised GANs.

5 Conclusion

In this paper, we present a novel method SemiStarGAN, which utilize unlabeled data for the problem of multi-domain image-to-image translation. Experimental

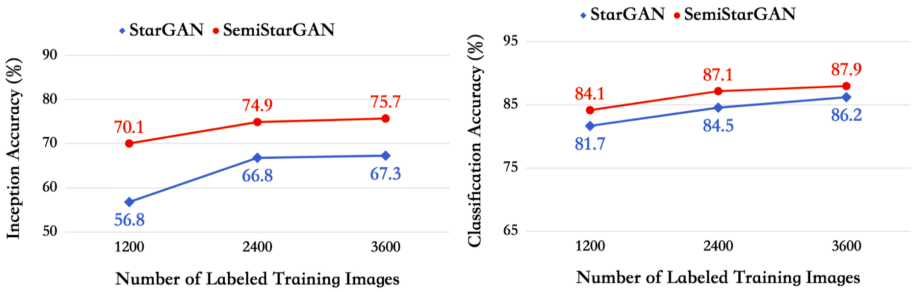

Fig. 8: Learning curves of StarGAN and SemiStarGAN of the facial attribute synthesis experiments using 12 domains. Images of both test and labeled training sets are evenly sampled from the 12 domains.

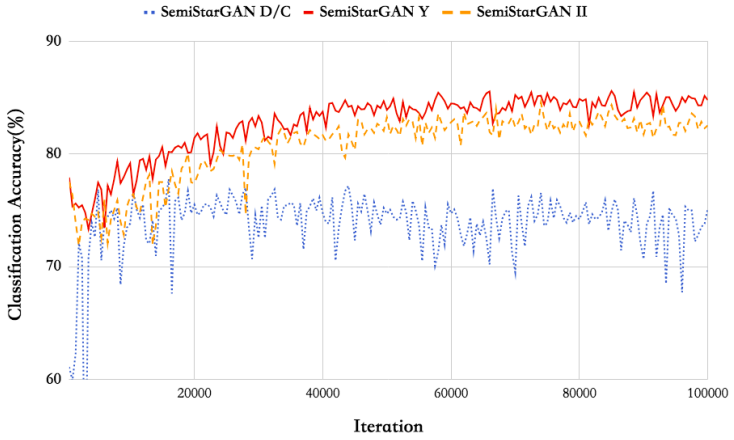

Fig. 9: Performance comparison of three different architecture models applicable to the discriminator and auxiliary classifier. D/C stands for the mixed model similar to the architecture used in StarGAN, and II stands for a model whose discriminator and classifier branches are totally separated.

results show the method's effectiveness for facial attribute transferring. For hair color transferring, SemiStarGAN only needs one third of the labeled data used by StarGAN to achieve the same Inception accuracy rate.

References

1. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR (2018)
2. Dai, Z., Yang, Z., Yang, F., Cohen, W.W., Salakhutdinov, R.: Good Semi-supervised Learning that Requires a Bad GAN. In: NIPS (2017)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)

4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
5. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: NIPS (2017)
6. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
7. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: ICML (2017)
8. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: ICML (2015)
9. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: ICLR (2017)
10. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
11. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning,ICML (2013)
12. Li, C., Xu, K., Zhu, J., Zhang, B.: Triple generative adversarial nets. In: NIPS (2017)
13. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: ECCV (2016)
14. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NIPS (2017)
15. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: NIPS (2016)
16. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015)
17. Miyato, T., Maeda, S.i., Koyama, M., Ishii, S.: Virtual Adversarial Training: a Regularization Method for Supervised and Semi-supervised Learning. ArXiv e-prints (Apr 2017)
18. Odena, A.: Semi-supervised learning with generative adversarial networks. In: workshop at ICML (2016)
19. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML (2016)
20. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS (2016)
21. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. In: ICLR (2016)
22. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
23. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: ICLR (2017)
24. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. ArXiv e-prints (Jul 2016)
25. Yi, Z., Zhang, H., Tan, P., Gong, M.: DualGAN: Unsupervised dual learning for image-to-image translation. In: ICCV (2017)
26. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
27. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017)

28. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: ECCV (2016)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: ICCV (2017)

SemiStarGAN: Semi-Supervised Generative Adversarial Networks for Multi-Domain Image-to-Image Translation - Appendix

Shu-Yu Hsu[1][0000−0003−4916−9926], Chih-Yuan Yang[1][0000−0002−8989−501X], Chi-Chia Huang[2][0000−0002−8517−5525], and Jane Yung-jen Hsu[1,2][0000−0002−2408−4603]

[1] Computer Science and Information Engineering, National Taiwan University
{r05922059,yangchihyuan,yjhsu}@ntu.edu.tw
[2] Graduate Institute of Networking and Multimedia, National Taiwan University
d01944003@ntu.edu.tw

1 Training Tips

We empirically train the proposed method using two tips. First, we alternately train the discriminator and generator in 6 consecutive and 1 iterations respectively. Second, we use an approach named Gaussian ramp up to adjust a weighting parameter λ_{cls}^{u} used in our self-ensembling loss $\mathcal{L}_{cls}^{u}$. We set the parameter as 0 at the beginning of the training process and gradually increase it until one-third of the training data are used. That is, we expect that the classifier loss primarily comes from labeled data in the early stage. Once the classifier reaches a certain level of precision, we let unlabeled data join the training process to reinforce the generalization ability of the classifier.

2 Network Architecture

Architecture of SemiStarGAN We show the network architecture of our generator in Table 1, discriminator in Table 2, and auxiliary classifier in Table 3. The conv_share_1 and conv_share_2 are the two layers shared by both the discriminator and auxiliary classifier. To apply self-ensembling, we put three dropout layers in the auxiliary classifier.

In order to match the requirement of output vector formats, we use two different ending layers in the auxiliary classifier network in our experiments. The ending layer is softmax in the three-domain experiment, and sigmoid in the twelve-domain one.

Since the three-domain problem is easier, we remove the conv_d_4 and conv_d_6 layers from the discriminator.

Architecture of Modified Inception-v3 Inception-v3 is originally designed for mutli-class object recognition and contains two softmax layers to generate

Table 1: Framework of our generator G. N: number of filters, K: kernel size, S: stride size, P: padding method, IN: instance norm.

Name	Description
input	128x128 RGB image
conv_downsample_1	N64, K7x7, P'Same', S1, ReLU, IN
conv_downsample_2	N128, K4x4, P'Same', S2, ReLU, IN
conv_downsample_3	N256, K4x4, P'Same', S2, ReLU, IN
conv_residual_1	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
conv_residual_2	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
conv_residual_3	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
conv_residual_4	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
conv_residual_5	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
conv_residual_6	Residual Block: N256, K3x3, P'Same', S1, ReLU, IN
deconv_upsample_1	128 filters, 4x4, pad= 'Same', stride=2, ReLU, instance norm
deconv_upsample_2	64 filters, 4x4, pad= 'Same', stride=2, ReLU, instance norm
deconv_upsample_3	3 filters, 7x7, pad= 'Same', stride=1, tanh

Table 2: Architecture of the discriminator in the Y model. Note that both the classifier and the discriminator share the first two layers(conv_share_1 and conv_share_2). N: the number of filters, K: kernel size, S: stride size, P: padding method.

Name	Description
input	128x128 RGB image
noise	Additive Gaussian noise, = 0.15
conv_share_1	N64, K4x4, P'Same', S2, LReLU(=0.1)
conv_share_2	N128, K4x4, P'Same', S2, LReLU(=0.1)
conv_d_3	N256, K4x4, P'Same', S2, LReLU(=0.1)
conv_d_4	N256, K4x4, P'Same', S2, LReLU(=0.1)
conv_d_5	N512, K4x4, P'Same', S2, LReLU(=0.1)
conv_d_6	N512, K4x4, P'Same', S2, LReLU(=0.1)
conv_d_patch	N1, K3x3, P'Same', S1

a one-dimension output vector. To incorporate Inception-v3 into our twelve-domain experiment which requires a three-dimension (hair color, gender, age) output vector to describe our domain labels, we replace Inception-v3's two softmax layers with sigmoid layers. We show the modified Inception-v3 architecture in Fig. 1.

Table 3: Architecture of our auxiliary classifier. Note that our classifier and the discriminator share the first two layers (conv_share_1 and conv_share_2). N: number of filters. K: kernel size, S: stride size, P: padding method, BN: batch norm, n_d: number of domains.

Name	Description
input	128x128 RGB image
noise	Additive Gaussian noise, = 0.15
conv_share_1	N64, K4x4, P'Same', S2, LReLU(=0.1)
conv_share_2	N128, K4x4, P'Same', S2, LReLU(=0.1)
drop1	Dropout (rate = 0.5)
conv_c_3	N256, K4x4, P'Same', S2, LReLU(=0.1), BN
conv_c_4	N256, K4x4, P'Same', S2, LReLU(=0.1), BN
drop2	Dropout (rate = 0.5)
conv_c_5	N512, K4x4, P'Same', S2, LReLU(=0.1), BN
conv_c_6	N512, K4x4, P'Same', S2, LReLU(=0.1), BN
drop3	Dropout (rate = 0.5)
conv_c_7	N256, K1x1, P'Same', S1, LReLU(=0.1), BN
conv_c_8	N128, K1x1, P'Same', S1, LReLU(=0.1), BN
pool1	Global average pool (2x2 to 1x1 pixel)
conv_c_logits	N(n_d), K1x1, P'Same', S1
output	Softmax (Sigmoid)

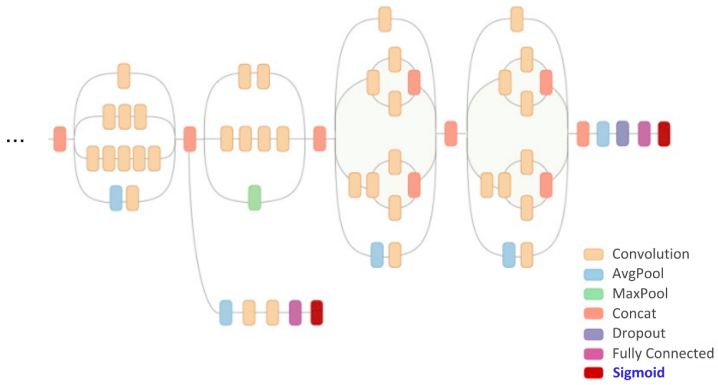

Fig. 1: The modified Inception-v3. Note the softmax layers are replaced by sigmoid (red blocks).

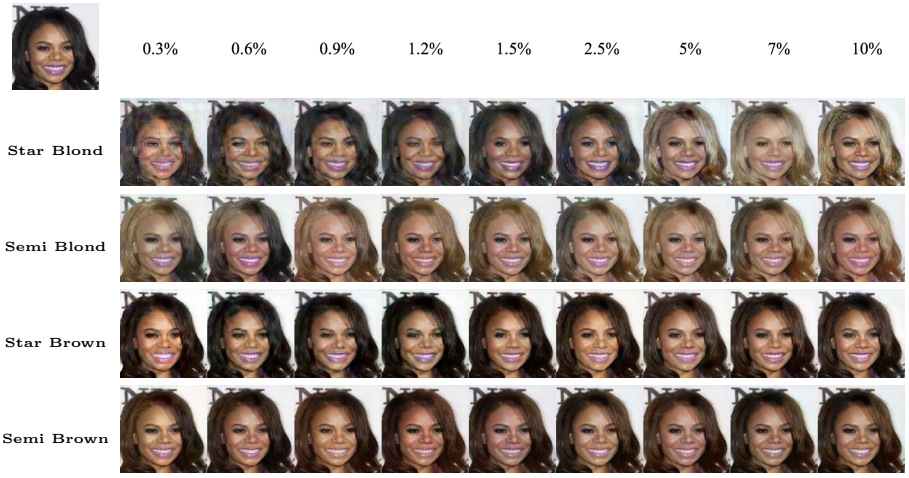

Fig. 2: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over three hair color domains. (Top-left) A source image belonging to the domain of black hair. (Top) Proportions of labeled images over all training images of the unit of percentage. (Left) The methods and their target domains. From the top two rows of images, the proposed method generates blond hair more obvious than the compared method using the same labeled training images under the proportions from 0.3% to 2.5%.

Fig. 3: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over three hair color domains. (Top-left) A source image belonging to the domain of brown hair. (Top) Proportions of labeled images over all training images of the unit of percentage. (Left) The methods and their target domains. From the top two rows of images, the proposed method generates black hair more obvious than the compared method using the same labeled training images under the proportions from 0.3% to 1.2%.

Fig. 4: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over three hair color domains. (Top-left) A source image belonging to the domain of blond hair. (Top) Proportions of labeled images over all training images of the unit of percentage. (Left) The methods and their target domains. Using the same training images, the proposed method generates not only better image quality in terms of less artifacts, but also more obvious image attributes such as black and brown hair on the second and fourth rows of images.

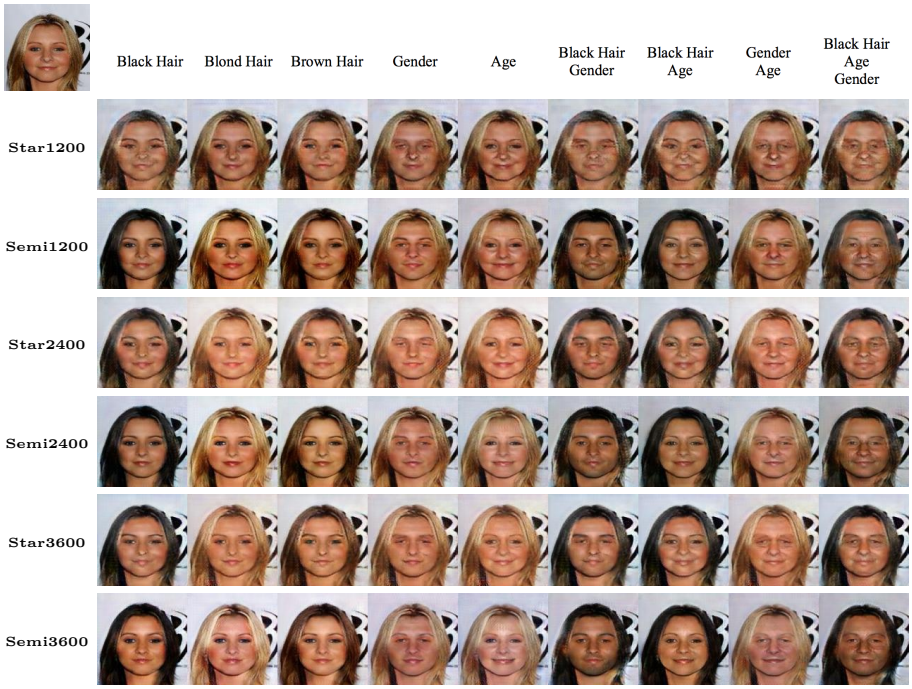

Fig. 5: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over 12-domain problem. (Top-left) A source image belonging to the domain of blond young female. (Top) The changed facial attributes. (Left) The methods and their used numbers of labeled training images. SemiStarGAN generates obvious black hair using 1200 labeled data but StarGAN can not generate the same level of salience even using 3600 labeled data. While transferring gender and age, SemiStarGAN generates clearer eyes and smoother skin than StarGAN.

Fig. 6: Qualitative comparison between the proposed method (Semi) and a state-of-the-art method StarGAN (Star) over 12-domain problem. (Top-left) A source image belonging to the domain of brown hair young female. (Top) The changed facial attributes. (Left) The methods and their used numbers of labeled training images. SemiStarGAN generates obvious black hair and blond hair using 1200 labeled data but StarGAN can not generate the same level of salience even using 2400 labeled data. While transferring gender and age, SemiStarGAN generates clearer eyes and smoother skin than StarGAN.