StarGAN v2: Diverse Image Synthesis for Multiple Domains

Yunjey Choi*   Youngjung Uh*   Jaejun Yoo*   Jung-Woo Ha

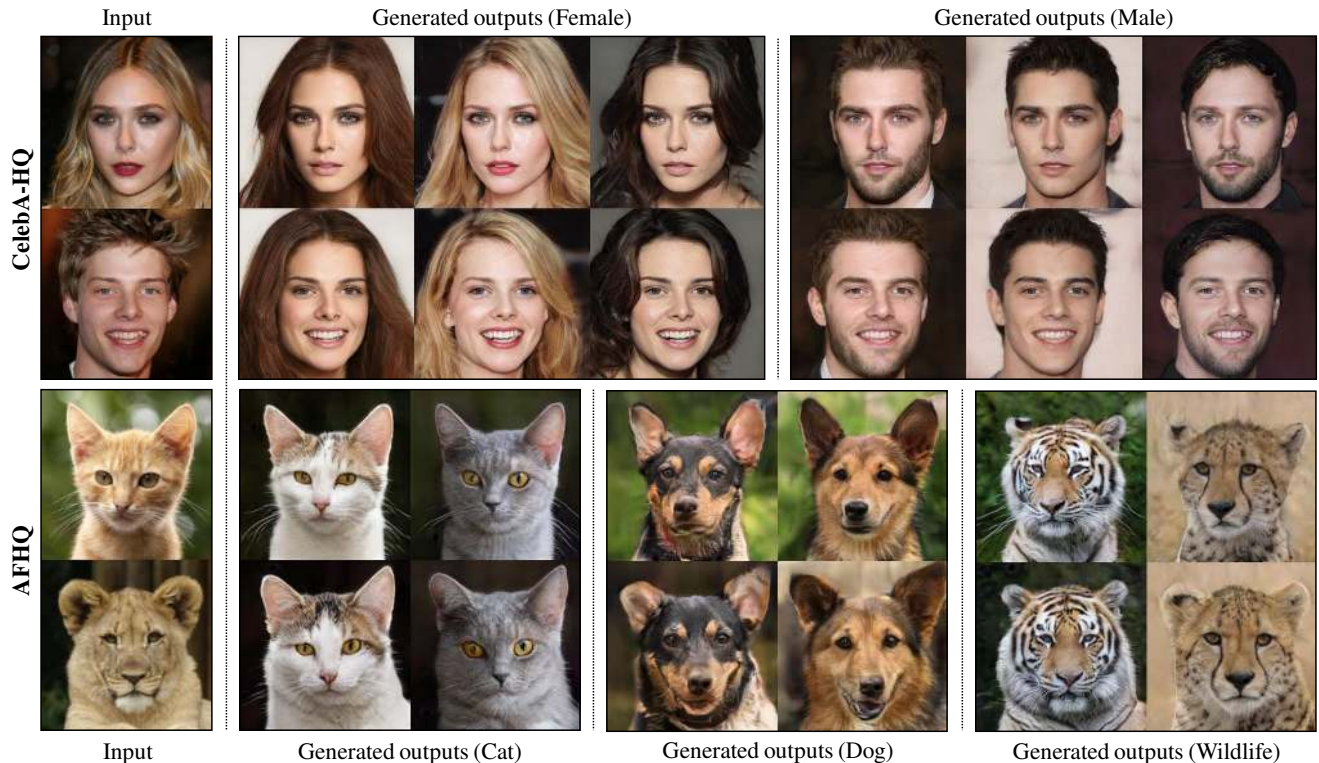Clova AI Research, NAVER Corp.

Figure 1. Diverse image synthesis results on the CelebA-HQ dataset and the newly collected animal faces (AFHQ) dataset. The first column shows input images while the remaining columns are images synthesized by StarGAN v2.

Abstract

A good image-to-image translation model should learn a mapping between different visual domains while satisfying the following properties: 1) diversity of generated images and 2) scalability over multiple domains. Existing methods address either of the issues, having limited diversity or multiple models for all domains. We propose StarGAN v2, a single framework that tackles both and shows significantly improved results over the baselines. Experiments on CelebA-HQ and a new animal faces dataset (AFHQ) validate our superiority in terms of visual quality, diversity, and scalability. To better assess image-to-image translation models, we release AFHQ, high-quality animal faces with large inter- and intra-domain differences. The code, pretrained models, and dataset are available at clovaai/stargan-v2.

1. Introduction

Image-to-image translation aims to learn a mapping between different visual domains [16]. Here, domain implies a set of images that can be grouped as a visually distinctive category, and each image has a unique appearance, which we call style. For example, we can set image domains based on the gender of a person, in which case the style includes makeup, beard, and hairstyle (top half of Figure 1). An ideal image-to-image translation method should be able to synthesize images considering the diverse styles in each domain. However, designing and learning such models become complicated as there can be arbitrarily large number of styles and domains in the dataset.

_____

* indicates equal contribution

To address the style diversity, much work on image-to-image translation has been developed [1, 13, 27, 22, 30, 40]. These methods inject a low-dimensional latent code to the generator, which can be randomly sampled from the standard Gaussian distribution. Their domain-specific decoders interpret the latent codes as recipes for various styles when generating images. However, because these methods have only considered a mapping between two domains, they are not scalable to the increasing number of domains. For example, having `K` domains, these methods require to train `K(K-1)` generators to handle translations between each and every domain, limiting their practical usage.

To address the scalability, several studies have proposed a unified framework [2, 6, 14, 24]. StarGAN [6] is one of the earliest models, which learns the mappings between all available domains using a single generator. The generator takes a domain label as an additional input, and learns to transform an image into the corresponding domain. However, StarGAN still learns a deterministic mapping per each domain, which does not capture the multi-modal nature of the data distribution. This limitation comes from the fact that each domain is indicated by a predetermined label. Note that the generator receives a fixed label (*e.g.* one-hot vector) as input, and thus it inevitably produces the same output per each domain, given a source image.

To get the best of both worlds, we propose StarGAN v2, a scalable approach that can generate **diverse** images across **multiple domains**. In particular, we start from StarGAN and replace its domain label with our proposed domain-specific style code that can represent diverse `styles` of a specific `domain`. To this end, we introduce two modules, a mapping network and a style encoder. The mapping network learns to transform random Gaussian noise into a style code, while the encoder learns to extract the style code from a given reference image. Considering multiple domains, both modules have multiple output branches, each of which provides style codes for a specific domain. Finally, utilizing these style codes, our generator learns to successfully synthesize diverse images over multiple domains (Figure 1).

We first investigate the effect of individual components of StarGAN v2 and show that our model indeed benefits from using the style code (Section 3.1). We empirically demonstrate that our proposed method is scalable to multiple domains and gives significantly better results in terms of visual quality and diversity compared to the leading methods (Section 3.2). Last but not least, we present a new dataset of animal faces (AFHQ) with high quality and wide variations (Appendix) to better evaluate the performance of image-to-image translation models on large inter- and intra-domain differences. We release this dataset publicly available for research community.

2. StarGAN v2

In this section, we describe our proposed framework and its training objective functions.

2.1. Proposed framework

Let $\mathcal{X}$ and $\mathcal{Y}$ be the sets of images and possible domains, respectively. Given an image $\mathbf{x} \in \mathcal{X}$ and an arbitrary domain $y \in \mathcal{Y}$, our goal is to train a **single** generator G that can generate **diverse** images of each domain y that corresponds to the image $\mathbf{x}$. We generate **domain-specific** style vectors in the learned style space of each domain and train G to reflect the style vectors. Figure 2 illustrates an overview of our framework, which consists of four modules described below.

Generator (Figure 2a). Our generator G translates an input image $\mathbf{x}$ into an output image $G(\mathbf{x}, \mathbf{s})$ reflecting a domain-specific style code $\mathbf{s}$, which is provided either by the mapping network F or by the style encoder E. We use adaptive instance normalization (AdaIN) [12, 18] to inject $\mathbf{s}$ into G. We observe that $\mathbf{s}$ is designed to represent a style of a specific domain y, which removes the necessity of providing y to G and allows G to synthesize images of all domains.

Mapping network (Figure 2b). Given a latent code $\mathbf{z}$ and a domain y, our mapping network F generates a style code $\mathbf{s} = F_y(\mathbf{z})$, where $F_y(\cdot)$ denotes an output of F corresponding to the domain y. F consists of an MLP with multiple output branches to provide style codes for all available domains. F can produce diverse style codes by sampling the latent vector $\mathbf{z} \in \mathcal{Z}$ and the domain $y \in \mathcal{Y}$ randomly. Our multi-task architecture allows F to efficiently and effectively learn style representations of all domains.

Style encoder (Figure 2c). Given an image $\mathbf{x}$ and its corresponding domain y, our encoder E extracts the style code $\mathbf{s} = E_y(\mathbf{x})$ of $\mathbf{x}$. Here, $E_y(\cdot)$ denotes the output of E corresponding to the domain y. Similar to F, our style encoder E benefits from the multi-task learning setup. E can produce diverse style codes using different reference images. This allows G to synthesize an output image reflecting the style $\mathbf{s}$ of a reference image $\mathbf{x}$.

Discriminator (Figure 2d). Our discriminator D is a multi-task discriminator [24, 28], which consists of multiple output branches. Each branch D_y learns a binary classification determining whether an image $\mathbf{x}$ is a real image of its domain y or a fake image $G(\mathbf{x}, \mathbf{s})$ produced by G.

2.2. Training objectives

Given an image $\mathbf{x} \in \mathcal{X}$ and its original domain $y \in \mathcal{Y}$, we train our framework using the following objectives.

Adversarial objective. During training, we sample a latent code $\mathbf{z} \in \mathcal{Z}$ and a target domain $\widetilde{y} \in \mathcal{Y}$ randomly, and
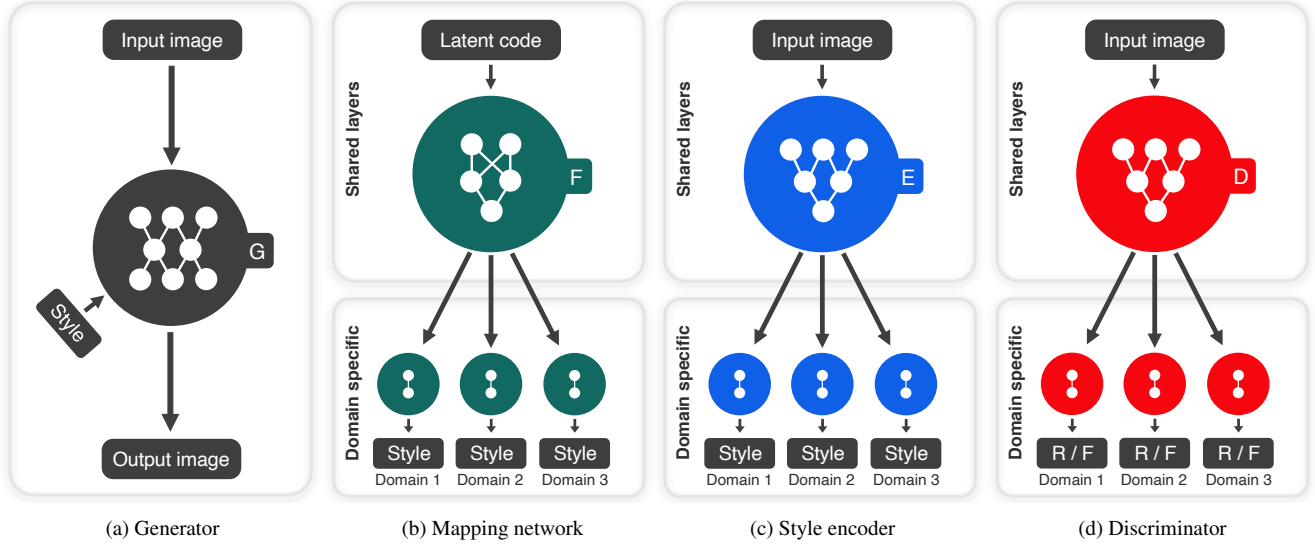
Figure 2. Overview of StarGAN v2, consisting of four modules. **(a)** The generator translates an input image into an output image reflecting the domain-specific style code. **(b)** The mapping network transforms a latent code into style codes for multiple domains, one of which is randomly selected during training. **(c)** The style encoder extracts the style code of an image, allowing the generator to perform reference-guided image synthesis. **(d)** The discriminator distinguishes between real and fake images from multiple domains. Note that all modules except the generator contain multiple output branches, one of which is selected when training the corresponding domain.

generate a target style code $\widetilde{s} = F_{\widetilde{y}}(\mathbf{z})$. The generator G takes an image $\mathbf{x}$ and $\widetilde{s}$ as inputs and learns to generate an output image $G(\mathbf{x}, \widetilde{s})$ via an adversarial loss

$$\begin{aligned}
\mathcal{L}_{adv} = {} & \mathbb{E}_{\mathbf{x},y}\left[\log D_y(\mathbf{x})\right] + \\
& \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}[\log\left(1 - D_{\widetilde{y}}(G(\mathbf{x},\widetilde{s}))\right)],
\end{aligned} \quad (1)$$

where $D_y(\cdot)$ denotes the output of D corresponding to the domain y. The mapping network F learns to provide the style code $\widetilde{s}$ that is likely in the target domain $\widetilde{y}$, and G learns to utilize $\widetilde{s}$ and generate an image $G(\mathbf{x},\widetilde{s})$ that is indistinguishable from real images of the domain $\widetilde{y}$.

Style reconstruction. In order to enforce the generator G to utilize the style code $\widetilde{s}$ when generating the image $G(\mathbf{x},\widetilde{s})$, we employ a style reconstruction loss

$$\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}}\left[||\widetilde{s} - E_{\widetilde{y}}(G(\mathbf{x},\widetilde{s}))||_1\right]. \quad (2)$$

This objective is similar to the previous approaches [13, 40], which employ multiple encoders to learn a mapping from an image to its latent code. The notable difference is that we train a **single** encoder E to encourage diverse outputs for multiple domains. At test time, our learned encoder E allows G to transform an input image, reflecting the style of a reference image.

Style diversification. To further enable the generator G to produce diverse images, we explicitly regularize G with the diversity sensitive loss [27, 35]

$$\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x},\widetilde{y},\mathbf{z}_1,\mathbf{z}_2}\left[||G(\mathbf{x},\widetilde{s}_1) - G(\mathbf{x},\widetilde{s}_2)||_1\right], \quad (3)$$

where the target style codes $\widetilde{s}_1$ and $\widetilde{s}_2$ are produced by F conditioned on two random latent codes $\mathbf{z}_1$ and $\mathbf{z}_2$ (*i.e.*

$\widetilde{s}_i = F_{\widetilde{y}}(\mathbf{z}_i)$ for $i \in \{1,2\}$). Maximizing the regularization term forces G to explore the image space and discover meaningful style features to generate diverse images. Note that in the original form, the small difference of $\|\mathbf{z}_1 - \mathbf{z}_2\|_1$ in the denominator increases the loss significantly, which makes the training unstable due to large gradients. Thus, we remove the denominator part and devise a new equation for stable training but with the same intuition.

Preserving source characteristics. To guarantee that the generated image $G(\mathbf{x},\widetilde{s})$ properly preserves the domain-invariant characteristics (*e.g.* pose) of its input image $\mathbf{x}$, we employ the cycle consistency loss [6, 20, 39]

$$\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x},y,\widetilde{y},\mathbf{z}}\left[||\mathbf{x} - G(G(\mathbf{x},\widetilde{s}),\hat{s})||_1\right], \quad (4)$$

where $\hat{s} = E_y(\mathbf{x})$ is the estimated style code of the input image $\mathbf{x}$, and y is the original domain of $\mathbf{x}$. By encouraging the generator G to reconstruct the input image $\mathbf{x}$ with the estimated style code $\hat{s}$, G learns to preserve the original characteristics of $\mathbf{x}$ while changing its style faithfully.

Full objective. Our full objective functions can be summarized as follows:

$$\begin{aligned}
\min_{G,F,E} \max_D \quad & \mathcal{L}_{adv} + \lambda_{sty}\,\mathcal{L}_{sty} \\
& - \lambda_{ds}\,\mathcal{L}_{ds} + \lambda_{cyc}\,\mathcal{L}_{cyc},
\end{aligned} \quad (5)$$

where λ_{sty}, λ_{ds}, and λ_{cyc} are hyperparameters for each term. We also further train our model in the same manner as the above objective, using reference images instead of latent vectors when generating style codes. We provide the training details in Appendix.

Method	FID	LPIPS
A Baseline StarGAN [6]	98.4	-
B + Multi-task discriminator	91.4	-
C + Tuning (*e.g.* R_1 regularization)	80.5	-
D + Latent code injection	32.3	0.312
E + Replace (D) with style code	17.1	0.405
F + Diversity regularization	**13.8**	**0.453**

Table 1. Performance of various configurations on CelebA-HQ. Frechet inception distance (FID) indicates the distance between two distributions of real and generated images (lower is better), while learned perceptual image patch similarity (LPIPS) measures the diversity of generated images (higher is better).

3. Experiments

In this section, we describe evaluation setups and conduct a set of experiments. We analyze the individual components of StarGAN v2 (Section 3.1) and compare our model with three leading baselines on diverse image synthesis (Section 3.2). All experiments are conducted using unseen images during the training phase.

Baselines. We use MUNIT [13], DRIT [22], and MSGAN [27] as our baselines, all of which learn multi-modal mappings between two domains. For multi-domain comparisons, we train these models multiple times for every pair of image domains. We also compare our method with StarGAN [6], which learns mappings among multiple domains using a single generator. All the baselines are trained using the implementations provided by the authors.

Datasets. We evaluate StarGAN v2 on CelebA-HQ [17] and our new AFHQ dataset (Appendix). We separate CelebA-HQ into two domains of male and female, and AFHQ into three domains of cat, dog, and wildlife. Other than the domain labels, we do not use any additional information (*e.g.* facial attributes of CelebA-HQ or breeds of AFHQ) and let the models learn such information as styles without supervision. For a fair comparison, all images are resized to 256×256 resolution for training, which is the highest resolution used in the baselines.

Evaluation metrics. We evaluate both the visual quality and the diversity of generated images using Frechét inception distance (FID) [11] and learned perceptual image patch similarity (LPIPS) [38]. We compute FID and LPIPS for every pair of image domains within a dataset and report their average values. The details on evaluation metrics and protocols are further described in Appendix.

3.1. Analysis of individual components

We evaluate individual components that are added to our baseline StarGAN using CelebA-HQ. Table 1 gives FID and LPIPS for several configurations, where each component is cumulatively added on top of StarGAN. An input image and the corresponding generated images of each configuration are shown in Figure 3. The baseline configura-

Figure 3. Visual comparison of generated images using each configuration in Table 1. Note that given a source image, the configurations (A) - (C) provide a single output, while (D) - (F) generate multiple output images.

tion (A) corresponds to the basic setup of StarGAN, which employs WGAN-GP [10], ACGAN discriminator [31], and depth-wise concatenation [29] for providing the target domain information to the generator. As shown in Figure 3a, the original StarGAN produces only a local change by applying makeup on the input image.

We first improve our baseline by replacing the ACGAN discriminator with a multi-task discriminator [28, 24], allowing the generator to transform the global structure of an input image as shown in configuration (B). Exploiting the recent advances in GANs, we further enhance the training stability and construct a new baseline (C) by applying R_1 regularization [28] and switching the depth-wise concatenation to adaptive instance normalization (AdaIN) [8, 12]. Note that we do not report LPIPS of these variations in Table 1, since they are yet to be designed to produce multiple outputs for a given input image and a target domain.

To induce diversity, one can think of directly giving a latent code $\mathbf{z}$ into the generator G and impose the latent reconstruction loss $||\mathbf{z} - E(G(\mathbf{x}, \mathbf{z}, y))||_1$ [13, 40]. However, in a *multi-domain* scenario, we observe that this baseline (D) does not encourage the network to learn meaningful styles and fails to provide as much diversity as we expect. We conjecture that this is because latent codes have no capability in separating domains, and thus the latent reconstruction loss models *domain-shared* styles (*e.g.* color) rather than *domain-specific* ones (*e.g.* hairstyle). Note that the FID gap between baseline (C) and (D) is simply due to the difference in the number of output samples.

Figure 4. Reference-guided image synthesis results on CelebA-HQ. The source and reference images in the first row and the first column are real images, while the rest are images generated by our proposed model, StarGAN v2. Our model learns to transform a source image reflecting the style of a given reference image. High-level semantics such as hairstyle, makeup, beard and age are followed from the reference images, while the pose and identity of the source images are preserved. Note that the images in each column share a single identity with different styles, and those in each row share a style with different identities.

| Source | MUNIT [13] | DRIT [22] | MSGAN [27] | **Ours** | Source | MUNIT [13] | DRIT [22] | MSGAN [27] | **Ours** |

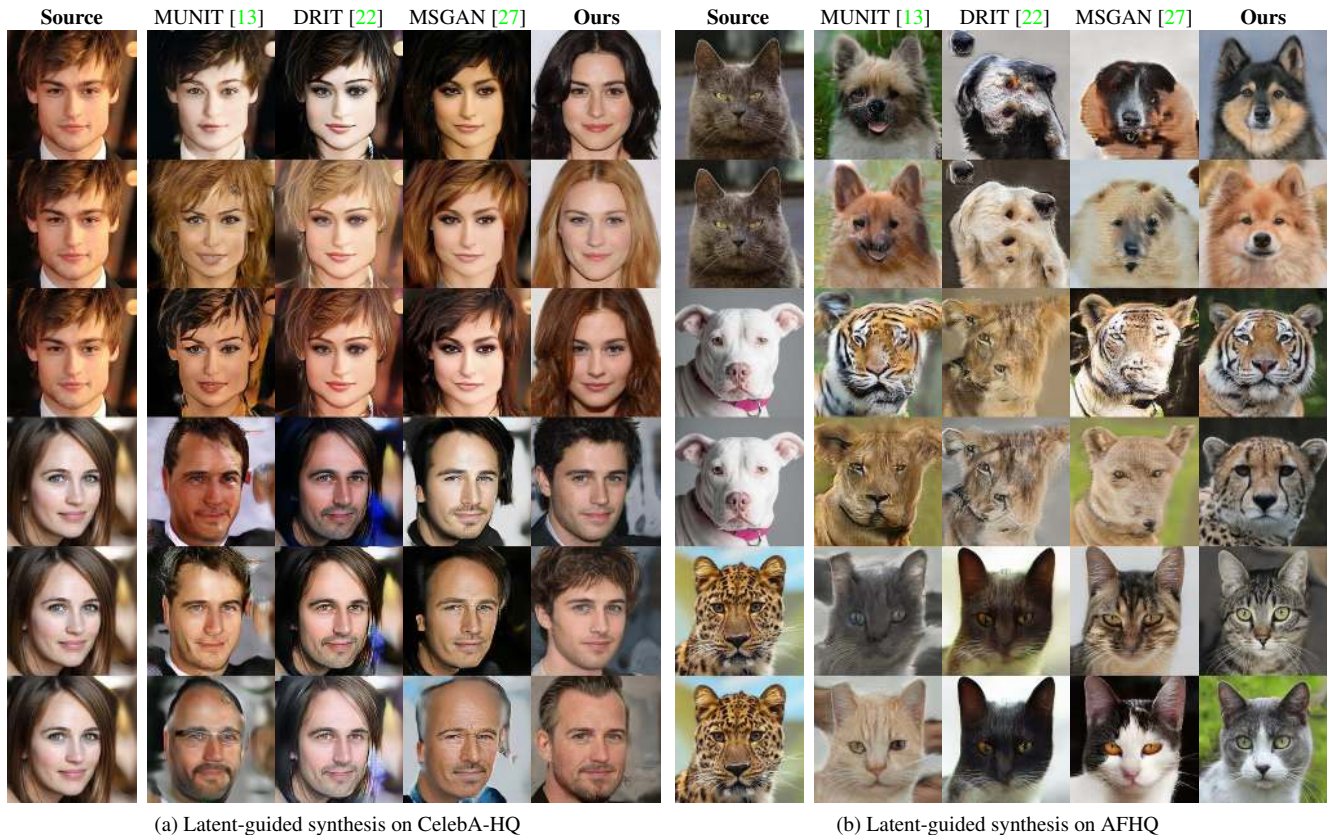(a) Latent-guided synthesis on CelebA-HQ (b) Latent-guided synthesis on AFHQ

Figure 5. Qualitative comparison of latent-guided image synthesis results on the CelebA-HQ and AFHQ datasets. Each method translates the source images (left-most column) to target domains using randomly sampled latent codes. **(a)** The top three rows correspond to the results of converting male to female and vice versa in the bottom three rows. **(b)** Every two rows from the top show the synthesized images in the following order: cat-to-dog, dog-to-wildlife, and wildlife-to-cat.

Instead of giving a latent code into G directly, to learn meaningful styles, we transform a latent code $\mathbf{z}$ into a *domain-specific* style code $\mathbf{s}$ through our proposed mapping network (Figure 2b) and inject the style code into the generator (E). Here, we also introduce the style reconstruction loss (Eq. (2)). Note that each output branch of our mapping network is responsible to a particular domain, thus style codes have no ambiguity in separating domains. Unlike the latent reconstruction loss, the style reconstruction loss allows the generator to produce diverse images reflecting *domain-specific* styles. Finally, we further improve the network to produce diverse outputs by adopting the diversity regularization (Eq. (3)), and this configuration (F) corresponds to our proposed method, StarGAN v2. Figure 4 shows that StarGAN v2 can synthesize images that reflect diverse styles of references including hairstyle, makeup, and beard, without hurting the source characteristics.

3.2. Comparison on diverse image synthesis

In this section, we evaluate StarGAN v2 on diverse image synthesis from two perspectives: latent-guided synthesis and reference-guided synthesis.
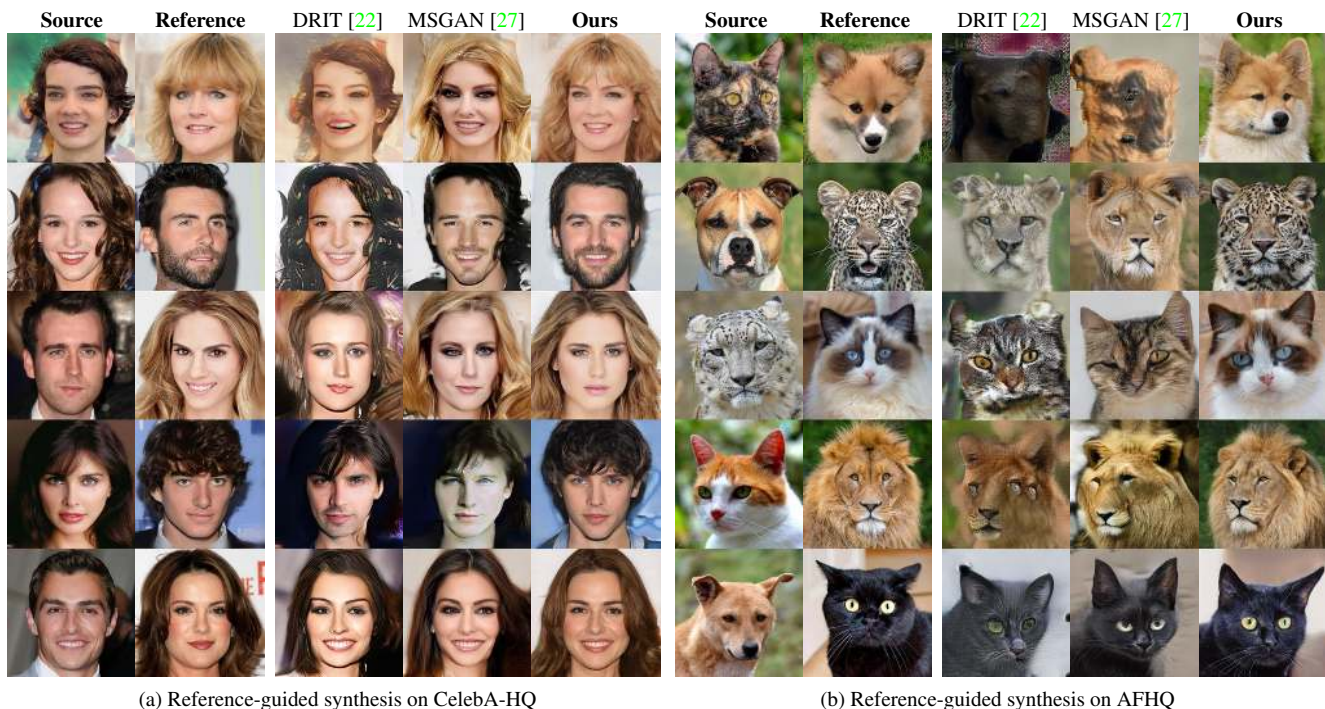
Latent-guided synthesis. Figure 5 provides a qualitative comparison of the competing methods. Each method pro-

Method	CelebA-HQ		AFHQ	
	FID	**LPIPS**	**FID**	**LPIPS**
MUNIT [13]	31.4	0.363	41.5	0.511
DRIT [22]	52.1	0.178	95.6	0.326
MSGAN [27]	33.1	0.389	61.4	**0.517**
StarGAN v2	**13.8**	**0.453**	**16.3**	0.451
Real images	14.8	-	12.9	-

Table 2. Quantitative comparison on latent-guided synthesis. The FIDs of real images are computed between the training and test sets. Note that they may not be optimal values since the number of test images is insufficient, but we report them for reference.

duces multiple outputs using random noise. For CelebA-HQ, we observe that our method synthesizes images with a higher visual quality compared to the baseline models. In addition, our method is the only model that can successfully change the entire hair styles of the source images, which requires non-trivial effort (*e.g.* generating ears). For AFHQ, which has relatively large variations, the performance of the baselines is considerably degraded, while our method still produces images with high quality and diverse styles.

As shown in Table 2, our method outperforms all the baselines by a large margin in terms of visual quality. For both CelebA-HQ and AFHQ, our method achieves FIDs of 13.8 and 16.3, respectively, which are more than two times

| Source | Reference | DRIT [22] | MSGAN [27] | **Ours** | Source | Reference | DRIT [22] | MSGAN [27] | **Ours** |

(a) Reference-guided synthesis on CelebA-HQ   (b) Reference-guided synthesis on AFHQ

Figure 6. Qualitative comparison of reference-guided image synthesis results on the CelebA-HQ and AFHQ datasets. Each method translates the source images into target domains, reflecting the styles of the reference images.

improvement over the previous leading method. Our LPIPS is also the highest in CelebA-HQ, which implies our model produces the most diverse results given a single input. We conjecture that the high LPIPS values of the baseline models in AFHQ are due to their spurious artifacts.

Reference-guided synthesis. To obtain the style code from a reference image, we sample test images from a target domain and feed them to the encoder network of each method. For CelebA-HQ (Figure 6a), our method successfully renders distinctive styles (*e.g.* bangs, beard, makeup, and hairstyle), while the others mostly match the color distribution of reference images. For the more challenging AFHQ (Figure 6b), the baseline models suffer from a large domain shift. They hardly reflect the style of each reference image and only match the domain. In contrast, our model renders distinctive styles (*e.g.* breeds) of each reference image as well as its fur pattern and eye color. Note that StarGAN v2 produces high quality images across all domains and these results are from a single generator. Since the other baselines are trained individually for each pair of domains, the output quality fluctuates across domains. For example, in AFHQ (Figure 6b), the baseline models work reasonably well in dog-to-wildlife (2nd row) while they fail in cat-to-dog (1st row).

Table 3 shows FID and LPIPS of each method for reference guided synthesis. For both datasets, our method achieves FID of 23.9, and 19.7, which are about 1.5× and 3.5× better than the previous leading method, respectively.

| | CelebA-HQ | | AFHQ | |
Method	FID	LPIPS	FID	LPIPS
MUNIT [13]	107.1	0.176	223.9	0.199
DRIT [22]	53.3	0.311	114.8	0.156
MSGAN [27]	39.6	0.312	69.8	0.375
StarGAN v2	**23.9**	**0.388**	**19.7**	**0.432**
Real images	14.8	-	12.9	-

Table 3. Quantitative comparison on reference-guided synthesis. We sample ten reference images to synthesize diverse images.

The LPIPS of StarGAN v2 is also the highest among the competitors, which implies that our model produces the most diverse results considering the styles of reference images. Here, MUNIT and DRIT suffer from mode-collapse in AFHQ, which results in lower LPIPS and higher FID than other methods.

Human evaluation. We use the Amazon Mechanical Turk (AMT) to compare the user preferences of our method with baseline approaches. Given a pair of source and reference images, the AMT workers are instructed to select one among four image candidates from the methods, whose order is randomly shuffled. We ask separately which model offers the best image quality and which model best stylizes the input image considering the reference image. For each comparison, we randomly generate 100 questions, and each question is answered by 10 workers. We also ask each worker a few simple questions to detect unworthy workers. The number of total valid workers is 76. As shown in Table 4, our method obtains the majority of votes in all in-

Method	CelebA-HQ		AFHQ	
	Quality	Style	Quality	Style
MUNIT [13]	6.2	7.4	1.6	0.2
DRIT [22]	11.4	7.6	4.1	2.8
MSGAN [27]	13.5	10.1	6.2	4.9
StarGAN v2	**68.9**	**74.8**	**88.1**	**92.1**

Table 4. Votes from AMT workers for the most preferred method regarding visual quality and style reflection (%). StarGAN v2 outperforms the baselines with remarkable margins in all aspects.

stances, especially in the challenging AFHQ dataset and the question about style reflection. These results show that StarGAN v2 better extracts and renders the styles onto the input image than the other baselines.

4. Discussion

We discuss several reasons why StarGAN v2 can successfully synthesize images of diverse styles over multiple domains. First, our style code is separately generated per domain by the multi-head mapping network and style encoder. By doing so, our generator can only focus on using the style code, whose domain-specific information is already taken care of by the mapping network (Section 3.1). Second, following the insight of StyleGAN [18], our style space is produced by learned transformations. This provides more flexibility to our model than the baselines [13, 22, 27], which assume that the style space is a fixed Gaussian distribution (Section 3.2). Last but not least, our modules benefit from fully exploiting training data from multiple domains. By design, the shared part of each module should learn domain-invariant features which induces the regularization effect, encouraging better generalization to unseen samples. To show that our model generalizes over the unseen images, we test a few samples from FFHQ [18] with our model trained on CelebA-HQ (Figure 7). Here, StarGAN v2 successfully captures styles of references and renders these styles correctly to the source images.

5. Related work

Generative adversarial networks (GANs) [9] have shown impressive results in many computer vision tasks such as image synthesis [3, 25, 7], colorization [15, 36] and super-resolution [21, 34]. Along with improving the visual quality of generated images, their diversity also has been considered as an important objective which has been tackled by either devoted loss functions [27, 28] or architectural design [3, 18]. StyleGAN [18] introduces a non-linear mapping function that embeds an input latent code into an intermediate style space to better represent the factors of variation. However, this method requires non-trivial effort when transforming a real image, since its generator is not designed to take an image as input.

Early image-to-image translation methods [16, 39, 23] are well known to learn a deterministic mapping even with stochastic noise inputs. Several methods reinforce the con-
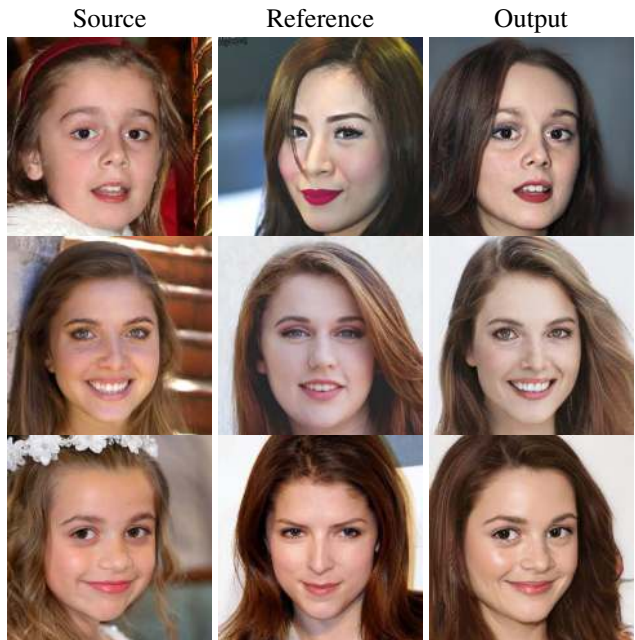

Figure 7. Reference-guided synthesis results on FFHQ with the model trained on CelebA-HQ. Despite the distribution gap between the two datasets, StarGAN v2 successfully extracts the style codes of the references and synthesizes faithful images.

nection between stochastic noise and the generated image for diversity, by marginal matching [1], latent regression [40, 13], and diversity regularization [35, 27]. Other approaches produce various outputs with the guidance of reference images [4, 5, 26, 32]. However, all theses methods consider only two domains, and their extension to multiple domains is non-trivial. Recently, FUNIT [24] tackles multi-domain image translation using a few reference images from a target domain, but it requires fine-grained class labels and can not generate images with random noise. Our method provides both latent-guided and reference-guided synthesis and can be trained with coarsely labeled dataset. In parallel work, Yu et al. [37] tackle the same issue but they define the style as domain-shared characteristics rather than domain-specific ones, which limits the output diversity.

6. Conclusion

We proposed StarGAN v2, which addresses two major challenges in image-to-image translation; translating an image of one domain to diverse images of a target domain, and supporting multiple target domains. The experimental results showed that our model can generate images with rich styles across multiple domains, remarkably outperforming the previous leading methods [13, 22, 27]. We also released a new dataset of animal faces (AFHQ) for evaluating methods in a large inter- and intra domain variation setting.

References

[1] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *ICML*, 2018. 2, 8

[2] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool. Combogan: Unrestrained scalability for image domain translation. In *CVPRW*, 2018. 2

[3] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 8

[4] H. Chang, J. Lu, F. Yu, and A. Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *CVPR*, 2018. 8

[5] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *CVPR*, 2019. 8

[6] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2, 3, 4

[7] J. Donahue and K. Simonyan. Large scale adversarial representation learning. In *NeurIPS*, 2019. 8

[8] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio. Feature-wise transformations. In *Distill*, 2018. 4

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 8

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 4

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 4

[12] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2, 4

[13] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2, 3, 4, 6, 7, 8

[14] L. Hui, X. Li, J. Chen, H. He, and J. Yang. Unsupervised multi-domain image translation with domain-specific encoders/decoders. In *ICPR*, 2018. 2

[15] K. Hyunsu, J. Ho Young, P. Eunhyeok, and Y. Sungjoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In *ICCV*, 2019. 8

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial nets. In *CVPR*, 2017. 1, 8

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 4

[18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 8

[19] H. Kim, M. Kim, D. Seo, J. Kim, H. Park, S. Park, H. Jo, K. Kim, Y. Yang, Y. Kim, et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 8

[20] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 3

[21] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 8

[22] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 2, 4, 6, 7, 8

[23] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 8

[24] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019. 2, 4, 8

[25] M. Lucic, M. Tschannen, M. Ritter, X. Zhai, O. Bachem, and S. Gelly. High-fidelity image generation with fewer labels. In *ICML*, 2019. 8

[26] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*, 2019. 8

[27] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019. 2, 3, 4, 6, 7, 8

[28] L. Mescheder, S. Nowozin, and A. Geiger. Which training methods for gans do actually converge? In *ICML*, 2018. 2, 4, 8

[29] M. Mirza and S. Osindero. Conditional generative adversarial nets. In *arXiv preprint*, 2014. 4

[30] S. Na, S. Yoo, and J. Choo. Miso: Mutual information loss with stochastic style representations for multimodal image-to-image translation. In *arXiv preprint*, 2019. 2

[31] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 4

[32] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 8

[33] N. Sung, M. Kim, H. Jo, Y. Yang, J. Kim, L. Lausen, Y. Kim, G. Lee, D. Kwak, J.-W. Ha, et al. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*, 2017. 8

[34] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV*, 2018. 8

[35] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2019. 3, 8

[36] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo. Coloring with limited data: Few-shot colorization via memory augmented networks. In *CVPR*, 2019. 8

[37] X. Yu, Y. Chen, T. Li, S. Liu, and G. Li. Multi-mapping image-to-image translation via learning disentanglement. In *NeurIPS*, 2019. 8

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*, 2017. 3, 8

[40] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 2, 3, 4, 8