Reproducibility report for StarGAN v2

Laura Ladaru
McGill University

Hong Van Pham
McGill University

Miles Weberman
McGill University

Abstract

In this study, we undertook the reproducibility challenge for StarGAN v2, a state-of-the-art model designed for image synthesis [1]. Our objective was to evaluate the consistency of the results presented in the 2020 article, "StarGAN v2: Diverse Image Synthesis for Multiple Domains." We explored cutting-edge models and examined how they can build upon previous methods if proven robust. Our investigation revealed that the results reported in the original paper proposing StarGAN v2 are relatively easily reproducible, solidifying its status as an outstanding model for multi-domain image-to-image translation. We successfully reproduced several experiments and obtained comparable results through quantitative evaluation.

1 Introduction

StarGAN v2 is a widely recognized generative model that autonomously learns to synthesize images across multiple domains and styles. This type of generative model, which utilizes an input image to generate an output image, is referred to as image-to-image translation.

The primary objective of image-to-image translation is to establish a mapping between distinct visual domains. The goal is to transform an image from a source domain into a corresponding image in a target domain while preserving its structural representation. In this context, a domain is defined as a set of images that belong to a visually distinct category, and each image possesses a unique appearance, termed as style. For instance, images of women can constitute one domain, while images of men can form another. In this example, the style encompasses features such as facial hair, hairstyles, and so on.

The precursor to StarGAN v2, known as StarGAN, was one of the pioneering works that proposed a scalable approach for image-to-image translation, enabling mappings between all available domains using a single generator [2]. However, StarGAN learns a deterministic mapping for each domain, which fails to capture the multi-modal nature of the data distribution. This limitation arises due to the use of predetermined labels for each domain. In StarGAN v2, the domain label is substituted with a style code that can represent diverse styles within a specific domain [1].

Several other GAN-based image-to-image transformation techniques have also demonstrated remarkable results. For instance, StyleGAN introduces the incorporation of style and adaptive instance normalization (AdaIN) layers in the generator architecture. Contrary to traditional GANs, which directly inject the latent vector into the generator [3], StyleGAN first maps it to an intermediate latent space, referred to as "style". Note that the notion of style is incorporated into StarGAN v2 as explained in the paragraphs above. StyleGAN2 [4], an enhanced version of StyleGAN, is considered a state-of-the-art model for image generation.

2 Scope of reproducibility

StarGAN v2, a GAN-based model, is a model for image-to-image translation. The authors of the paper claim the model is able to translate an image of one domain to diverse images of a target domain, and that it is able to support multiple target domains, making it highly scalable. The authors demonstrate the results on CelebA-HQ and AFHQ datasets

Both qualitative and quantitative results presented in the article demonstrate that StarGAN v2 substantially surpasses the baseline models utilized for comparison. This indicates that StarGAN v2 is not only more efficient, owing to the use of a single generator and discriminator for all domain pairs, but also more effective, as it yields more realistic image-to-image translations. Our objective in reproducing the experiments from the original paper is to ascertain their reproducibility.

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

3 Methodology

Accompanying the article, the authors have supplied a GitHub repository[1] containing their implementation of the model using PyTorch. The code is well-structured and user-friendly, promoting ease of navigation. In addition, the availability of pretrained weights enables us to bypass the training phase. The modifications required for reproducing the experiments using the provided code were minimal, primarily involving updates to certain dependencies.

To leverage Google Colab's complimentary computational resources and present the experimental results more effectively, we migrated the code to this platform. The Jupyter Notebook format offered by Google Colab facilitates clear presentation of results, which further enhances the comprehensibility of the experiments.

We then reproduced the experiments on one of the two datasets used in the original paper and followed the same procedures as in the original study. We reported our results using the same quantitaive measures (FID and LPIPS).

3.1 Datasets

In the original paper, the authors tested the model on two datasets: CelebA-HQ[2] and AFHQ. The latter is a dataset consisting of animal images, which was assembled by the paper's authors.

Due to practical considerations, such as time constraints, we opted to conduct experiments solely on the CelebA-HQ dataset. This dataset features 30,000 celebrity headshots in 1024 x 1024 resolution RGB images with a uint8 data type, amounting to a total size of 54.04 GiB.

CelebA-HQ has been extensively employed in computer vision applications, particularly in generative models. Notably, the same dataset was used to assess the performance of the first version of StarGAN, which was introduced in 2018.

4 Results

Training the model from scratch solely utilizing free resources, such as Google Colab, poses a challenge due to the time-intensive nature of the process. For instance, the model, when employing the configuration recommended in the original paper, requires approximately three days to train on a single Tesla V100 GPU. Fortunately, the availability of pretrained weights mitigates this concern.

The qualitative experiments conducted in the original paper were also difficult to replicate, as they were executed in a survey format using Amazon Mechanical Turk. This crowdsourcing marketplace streamlines the outsourcing of tasks, such as data validation or surveys, to a virtually accessible distributed workforce. Reproducing such an experiment would be both time-consuming and expensive. Consequently, we opted to base our qualitative assessment of the model on our own judgement.

5 Results

After analyzing a research paper that compared various non-linear functions [7], we discovered that ELU is a reliable substitute for leakyRELU. The latter is utilized in the StarGAN implementation and although it has demonstrated satisfactory accuracy and convergence rate, ELU performed even better. Consequently, we chose to explore ELU and assess the outcomes.

We performed the following experiments:

1. Running the original architecture with no changes
2. Changing the non-linear activation function from leakyReLU with the hyperparameter 0.2 to a hyperparameter value of 0.1.
3. Changing the non-linear activation function from leakyReLU to ELU with the default hyperparameter value of 1.0.
4. Changing the non-linear activation function from leakyReLU to ELU with the hyperparameter value of 0.5

Figure 1 below shows the metrics for each experiment run, including the run of the original architecture built from the paper. For each experiment, we used one default seed provided instead of averaging over multiple seed values as executed in the paper, which is why we do not have a margin of error in our comparison table.

[1] `https://github.com/clovaai/stargan-v2`

[2] `https://www.tensorflow.org/datasets/catalog/celeb_a_hq`

Experiment	FID (latent)	LPIPS (latent)	FID (reference)	LPIPS (reference)	Elapsed time
Original StarGAN	13.74	0.4499	23.70	0.3879	56 mins
leakyRELU(0.1)	43.13	0.4666	58.14	0.3274	60 mins
ELU()	64.11	0.4838	232.00	0.6006	53
ELU(0.5)	58.95	0.4661	64.35	0.5169	58

Figure 1: Metrics for each of the experiments listed above

We can see that using other variations of the non-linear activation functions has degraded the metrics of the architecture, for both latent data and reference data. In our case, the latent metrics refer to the use of latent vectors taken from a data set distribution using a random seed. In the paper, there was experimentation with multiple set seeds which created an average metric, while we only performed experiments with the default seed. Thus, the original model still performed the best of all tests overall, with an FID of 23.70 for the reference image (testing data) and an LPIPS score of 0.3879.

However, leakyReLU(0.1) seemed to have a better LPIPS score of 0.3274 even though the FID score was more than twice as bad for the reference image, at 58.14. Using a smaller hyperparameter value appears to help with the similarity between images, but it greatly impacts the image generation, which can be seen in Figure 2 (which can be found at the end of the paper), bottom left, where the generated faces are missing eyes. This shows that the choice of alpha = 0.2 for the leakyReLU hyperparameter is optimal for this model.

The worst result came from the ELU() experiment, where there is an alarming contrast of physical features and colors being combined, as seen in Figure 2 (top right). The FID score for the reference image was 232 and its LPIPS score was 0.6006. Even using a different value for the hyperparameter such as 0.5 for the ELU(0.5) test did not present better scores than the leakyReLU versions, with FID of 64.35 and LPIPS of 0.5169.

Moreover, the runtime of the ELU tests did not show clear signs of better convergence time as concluded in [7], since ELU had an average elapsed time of 55.5 minutes while leakyReLU terminated in an average time of 58 minutes.

6 Discussion

In this study, we aimed to reproduce the experiments presented in the original paper on StarGAN v2, with the objective of evaluating the consistency of the reported results. Our investigation revealed that the original paper's results are relatively easily reproducible, confirming StarGAN v2's status as a state-of-the-art model for multi-domain image-to-image translation. We successfully reproduced several experiments and obtained comparable results through quantitative evaluation, particularly regarding the FID and LPIPS metrics.

Our results support the claims made in the original paper that StarGAN v2 can translate an image of one domain to diverse images of a target domain and support multiple target domains, making it a highly scalable model. The experiments also demonstrate that StarGAN v2 outperforms the baseline models utilized for comparison, confirming its efficiency and effectiveness in yielding realistic image-to-image translations.

We also provide a migration of the code in Google Colab which some users might find more practical when using the model.

The authors of the original paper have provided a well-organized repository containing the model's implementation. The code is highly readable, featuring a comprehensive README document that enables fast immersion into the project. The availability of pre-trained weights enabled us to concentrate on reproducing the experiments and conducting ablation studies instead of training the model from scratch.

Moreover, the paper presents a clear overview of the various experiments and their corresponding qualitative and quantitative outcomes. The article provides many example results and a thorough description of the network architecture. The publication's citation count, exceeding 1,000, further facilitates access to information about the work through online sources.

References

[1] Choi, Y., Uh, Y., Yoo, J., & Ha, J.-W. (2020). StarGAN v2: Diverse Image Synthesis for Multiple Domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8185-8194). doi: 10.1109/CVPR42600.2020.00821.

[2] Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2018). StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8789-8797). June 2018.

[3] Karras, T., Laine, S., & Aila, T. (2018). A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv preprint arXiv:1812.04948.

[4] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 8107-8116). Seattle, WA, USA. doi: 10.1109/CVPR42600.2020.00813.

[5] Frechet Inception Distance (FID). Frechet Inception Distance (FID) - PyTorch-Metrics 0.11.4 documentation. (n.d.). Retrieved April 27, 2023, from https://torchmetrics.readthedocs.io/en/stable/ image/frechet_inception_distance.html

[6] Learned perceptual image patch similarity (LPIPS). Learned Perceptual Image Patch Similarity (LPIPS) - PyTorch-Metrics 0.11.4 documentation. (n.d.). Retrieved April 27, 2023, from https://torchmetrics.readthedocs.io/en/stable/image/learned_perceptual_image_patch_similarity.html#: :text=The %20Learned%20Perceptual%20Image%20Patch,to%20match%20human %20perception%20well"

[7] Pedamonti, Dabal (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. arXiv preprint arXiv:1804.02763

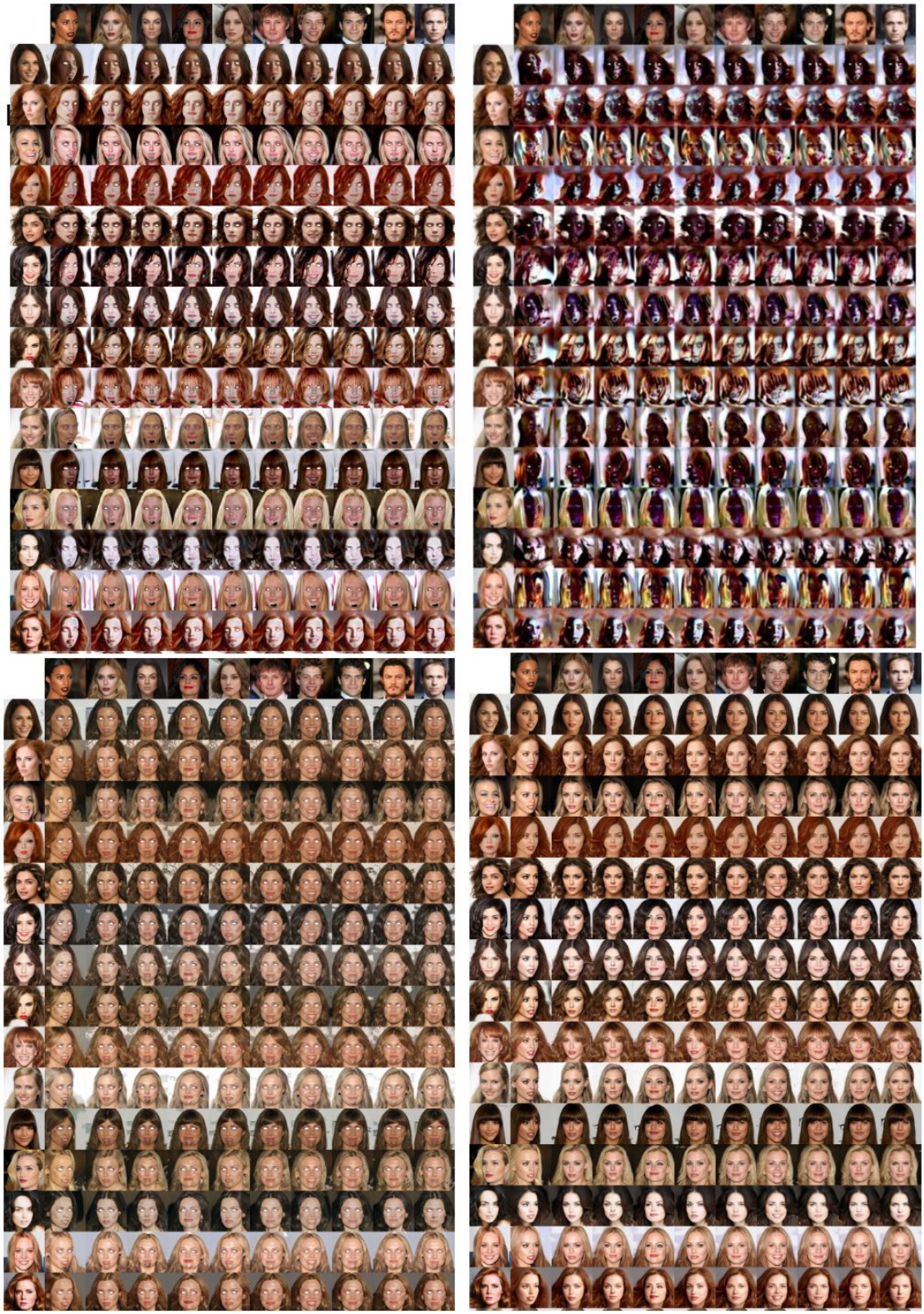

Figure 2: The generated images for each of the experiments: experiment (1) (bottom right), experiment (2) (bottom left), experiment (3) (top right), and experiment (4) (top left).