JoJoGAN: One Shot Face Stylization

Min Jin Chong and David Forsyth
University of Illinois at Urbana-Champaign
{mchong6, daf}@illinois.edu

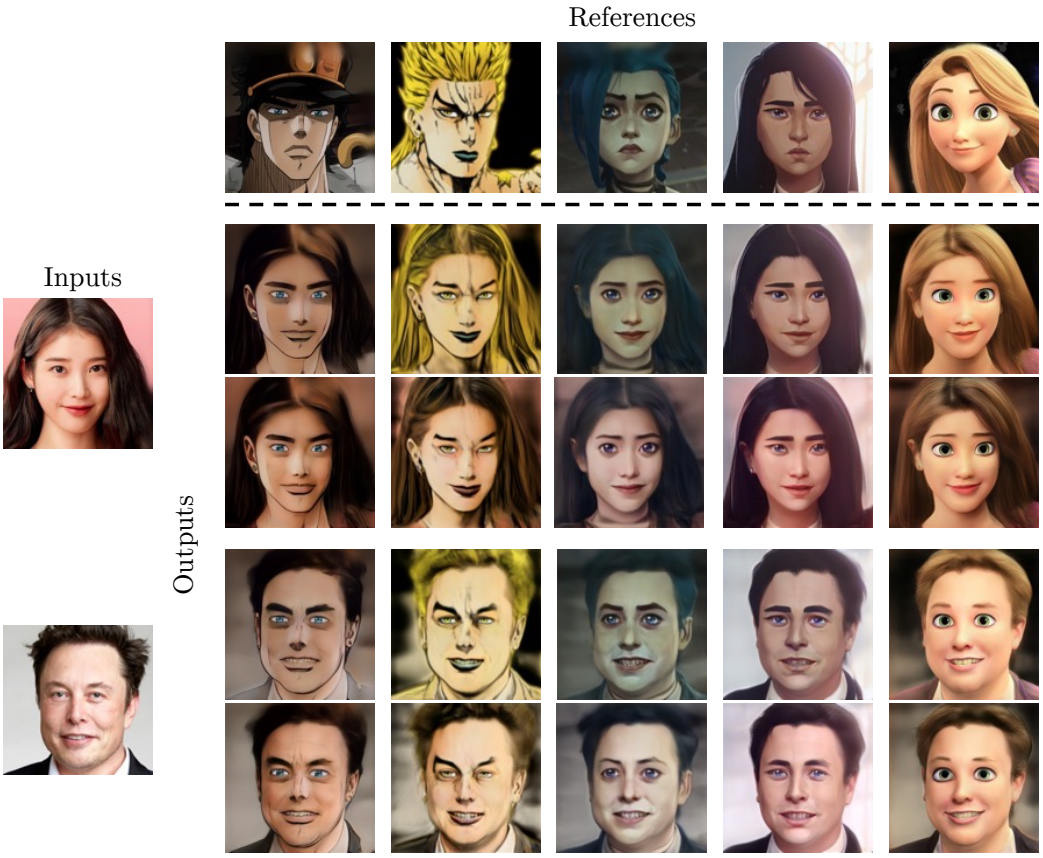


Figure 1: We perform arbitrary one-shot face stylization without any paired data. Only one single reference image is needed for training which takes about 1 minute. After training, style can be applied to any input image. We can selectively choose to preserve the original colors of the input images or transfer the colors from the reference style.

Abstract

While there have been recent advances in few-shot image stylization, these methods fail to capture stylistic details that are obvious to humans. Details such as the shape of the eyes, the boldness of the lines, are especially difficult for a model to learn, especially so under a limited data setting. In this work, we aim to perform one-shot image stylization that gets the details right. Given a reference style image, we approximate paired real data using GAN inversion and finetune a pretrained StyleGAN using that approximate paired data. We then encourage the StyleGAN to generalize so that the learned style can be applied to all other images.

1 Introduction

Stop spending your time collecting data! We present JoJoGAN (named after the best anime JoJo's Bizarre Adventure, the inspiration for this work), a framework for arbitrary one-shot face stylization. Given **a single input style reference**, we are able to apply the style onto any input image, carefully preserving detailed style characteristics such as eye appearences, proportions, etc.

Given only a single reference style image, a skilled artist can reproduce new artworks that faithfully capture the style. This is however, something that remains difficult with our current machine learning frameworks. The best way to train an image translation framework is with paired training data. This is however not realistic under real world setting. Oftentimes we are not able to get a large amount of training data, let alone paired data. Thus, much emphasis is placed on unpaired image to image translation that significantly reduces this constraint. Still, large amounts of data is needed, ranging from hundreds to thousands. While there have been some work on few shot translation [1, 2, 3], the results fail to capture distinct style details, diversity, or lack image quality.

JoJoGAN aims to solve this problem by first approximating a paired training dataset and then finetuning a StyleGAN to perform one-shot face stylization. We show that our method pays close attention to style details with zero supervision and generalizes well across various different styles. Training and demo code is available at https://github.com/mchong6/JoJoGAN.

2 Methodology

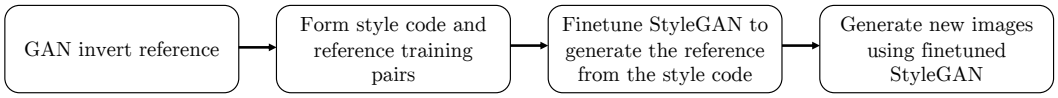


Figure 2: **Workflow**

JoJoGAN works by finetuning a pretrained StyleGAN2 [4] with a single reference style image. There are several steps in the pipeline,

1. We prepare approximate paired training data by GAN inverting the reference style image y, giving us style code w that generates a plausible corresponding real face image x.
2. We then find a family of $\mathcal{W}$ that generates a family of real face images $\mathcal{X}$ that should match reference style image y. Form pairs of (w_i, y) that serves as our paired training set.
3. Finetune the StyleGAN based on those paired training data.
4. Generate new samples using the finetuned StyleGAN.

2.1 Data Preparation

Training with paired data is optimal for the image stylization task. However, they are oftentimes very difficult to come by, requiring significant time and money investment. No good open-source paired dataset for our task is currently available. We aim to overcome this problem by generating an approximate paired training dataset as shown in Figure 3. Given a style reference image y, we perform GAN inversion using e4e [5] to obtain w. As e4e is trained on real face images, it fails to generalize to our out-of-distribution style image and thus giving us a w that approximates the "real" face image of y, forming a paired (w, y) training set.

Training using only a single datapoint leads to poor generalization to other images, see Figure 4. We overcome this by generating more training datapoints. The idea is simple, many real face images should match to the same style reference image. For example, faces with slightly different eye sizes or hair texture can reasonably be matched to the same reference image. It is trivial to generate these similar samples in StyleGAN by perform style mixing at certain chosen layers. For a 1024 resolution StyleGAN2 with 18 style modulation layers, our style code is $w \in \mathbb{R}^{18\times512}$. We define a mask $M \in \{0, 1\}^{18}$ which selectively masks out parts of our style code that we want to style mix for data augmentation, FC as the style mapping layer of the StyleGAN, $z_i \sim \mathcal{N}(0, I)$ as our random noise

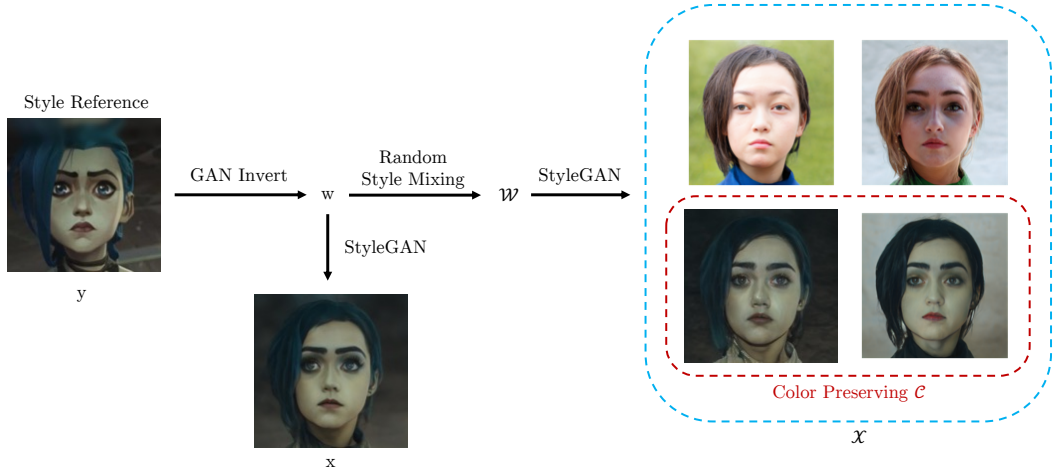


Figure 3: **Data Preparation:** We perform GAN inversion on the style reference image y to obtain a style code w for the corresponding "real" face image. We can then perform random style mixing to obtain a family of these style codes $\mathcal{W}$ whose face images $\mathcal{X}$ map to y, forming (w_i, y) paired training data. Among $\mathcal{X}$ is a subset of color preserving face images $\mathcal{C}$ that have similar color profile as y. Using $\mathcal{X}$ or $\mathcal{C}$ for our training data leads to different results, allowing us to control if our face stylization preserves the original color of the input image.

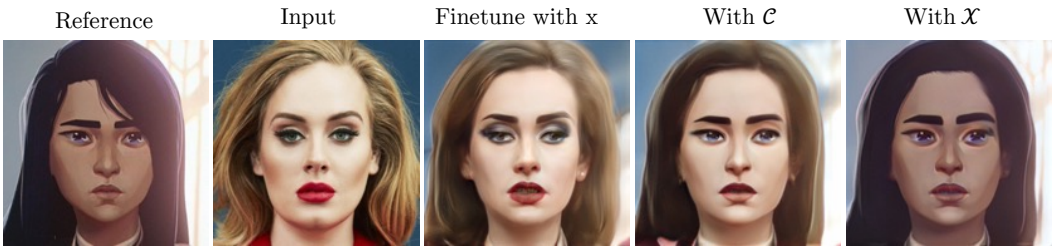


Figure 4: **Using different datasets:** We compare finetuning StyleGAN using different datasets. When finetuning with only a single datapoint x, our model is unable to fully capture the style details of the reference image and generalize to other images. For the dataset $\mathcal{C}$ which corresponds to face images with similar color profile to the reference image (refer to Figure 3), our finetuned model accurately captures style characteristics while keeping the originals colors of the input image. Finetuning with $\mathcal{X}$ instead copies the colors from the reference image, fully reproducing the original style.

vector, and $\alpha \sim \mathcal{U}(0, 1)$ a random scalar value controlling strength of style mixing. Our new style code w_i is thus,

$$w_i = (1 - \alpha)M \cdot w + \alpha(1 - M) \cdot FC(z_i) \tag{1}$$

Our choice of M determines the layers we opt to do style mixing. This choice of layers allows us to generate a family of similar face images $\mathcal{X}$ and control the color profile of them. For example, we generate dataset $\mathcal{X}$ in Figure 3 by style mixing layers 7 to 18, preserving pose and hairstyle, while varying colors, eye shapes, etc. By style mixing 7 to 9, we can preserve the colors of the style image, giving us subset $\mathcal{C}$.

2.2 Finetuning

After generating paired data, we can finetune a pretrained StyleGAN $\mathcal{G}$. Let the training pairs be (w_i, y) where w_i are the style codes we obtain previously, and y is the reference style image. We then define our loss as

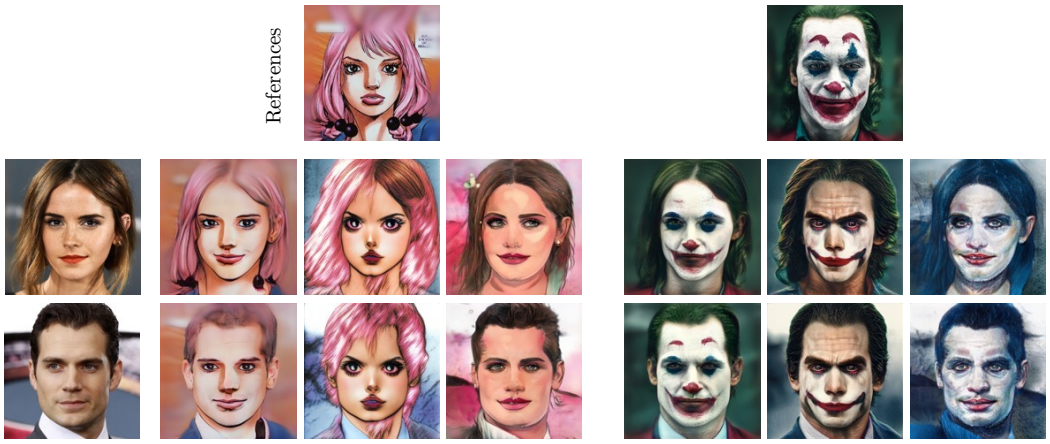


Figure 5: **Comparisons:** We compare our non-color preserving model with StyleGAN-NADA [8] and BlendGAN [2] on one shot face stylization. Our method captures details such as the hair accessories in (a) and the face paintings in (d). We also maintain the identity of the input image well, keeping expressions and hairstyles consistent. On the contrary, StyleGAN-NADA changes the facial identities and fails to capture the complex face paintings. BlendGAN fails to capture meaningful style features.

$$\text{loss} = \frac{1}{N}\sum_{i}^{N} \text{LPIPS}(\mathcal{G}(w_i), y) \tag{2}$$

using LPIPS [6] as our perceptual loss. Our different choices of training datasets lead to different results as seen in Figure 4. Unsurprisingly, training with only a single example x leads to poor results that fails to capture the proper style. As the training dataset $\mathcal{C}$ has the same color as the reference image, our finetuned StyleGAN learns to not drastically change the color of the images during finetuning process. This results in our finetuned model preserving the colors of the input image, not affected strongly by the style reference. Training with $\mathcal{X}$ instead fully captures the original style, including the colors.

2.3 Generation

Our fintuned StyleGAN now generates images with the reference style. To perform image translation, we simply perform GAN inversion on the input image and perform inference on our finetuned StyleGAN.

2.4 Multi-shot

Note that our method can easily extend to multi-shot stylization by simply minimizing Equation (2) w.r.t style codes corresponding to more style references.

3 Experiments

3.1 Setup

We finetune JoJoGAN for 500 iterations with Adam optimizer [7] at a learning rate of 2×10^{-3}. Finetuning on an Nvidia A40 takes about 1 minute. We compare non-color preserving JoJoGAN with state-of-the-art one/few shot stylization methods StyleGAN-NADA [8] and BlendGAN [2]. Compared to them, JoJoGAN can capture small details that define the style while maintaining clear facial identity from the inputs. In Figure 5(a), our method captures the eye shapes and details

perfectly and also the hair accessories from the style reference, while in Figure 5(d), we capture the complex face paintings accurately. In contrast, while StyleGAN-NADA captures the overall Joker makeup in Figure 5(e), it fails to capture the details such as the eyes and eyebrows paint. Identities are also majorly affected. BlendGAN fails to capture meaningful style details, with even hairstyles getting the wrong colors.

4 Future Work

While JoJoGAN allows simple one-shot face stylization, it is not practical to perform video inference due to the GAN inversion step. A straightforward way to perform inference is to first finetune our JoJoGAN and use it to produce new training dataset for real and stylized faces. We can then train a simple encoder decoder network in a supervised fashion and perform efficient inference on it.

References

[1] Utkarsh Ojha, Yijun Li, Cynthia Lu, Alexei A. Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *CVPR*, 2021. 2

[2] Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, and Wen Zheng. Blendgan: Implicitly gan blending for arbitrary stylized face generation. In *Advances in Neural Information Processing Systems*, 2021. 2, 4

[3] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsueprvised image-to-image translation. In *arxiv*, 2019. 2

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2

[5] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 2

[6] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 4

[8] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 4

5 Appendix

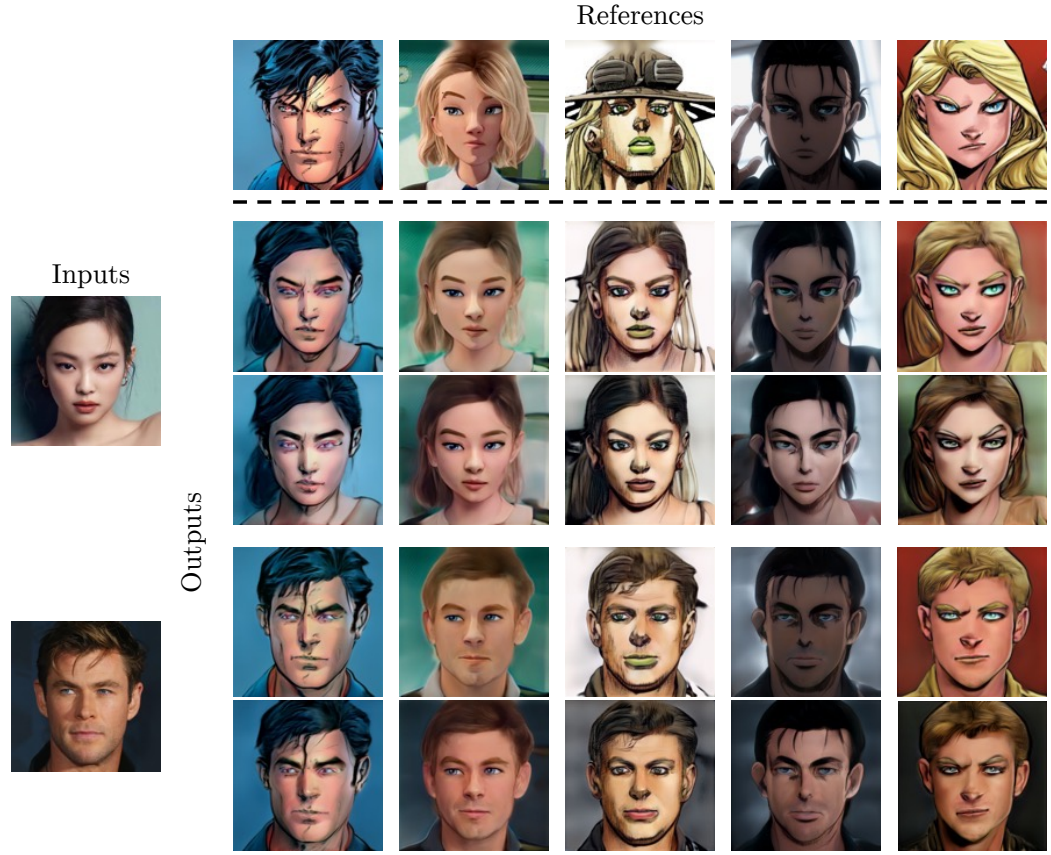
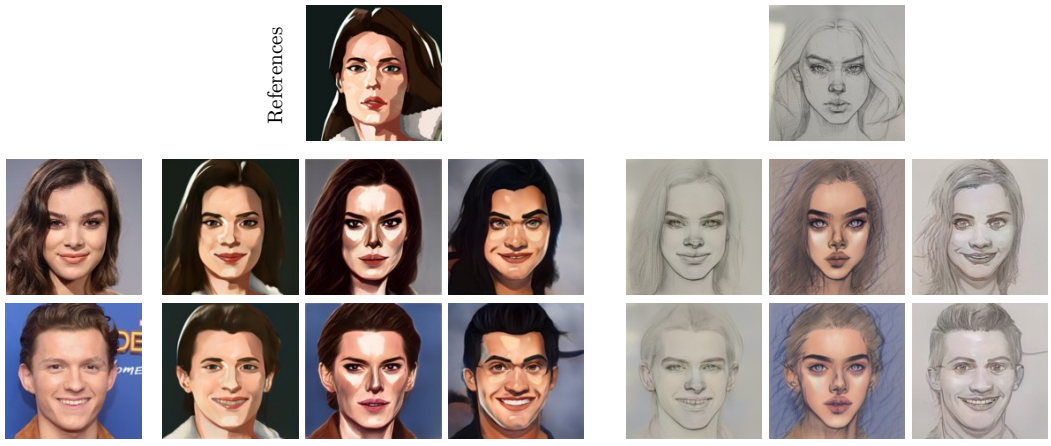


Figure 6: **More examples**



(a) JoJoGAN (b) StyleGAN -NADA (c) BlendGAN (d) JoJoGAN (e) StyleGAN -NADA (f) BlendGAN

Figure 7: **More comparisons**