JoJoGAN: One Shot Face Stylization

Min Jin Chong and D.A. Forsyth

University of Illinois at Urbana-Champaign
mchong6@illinois.edu daf@illinois.edu

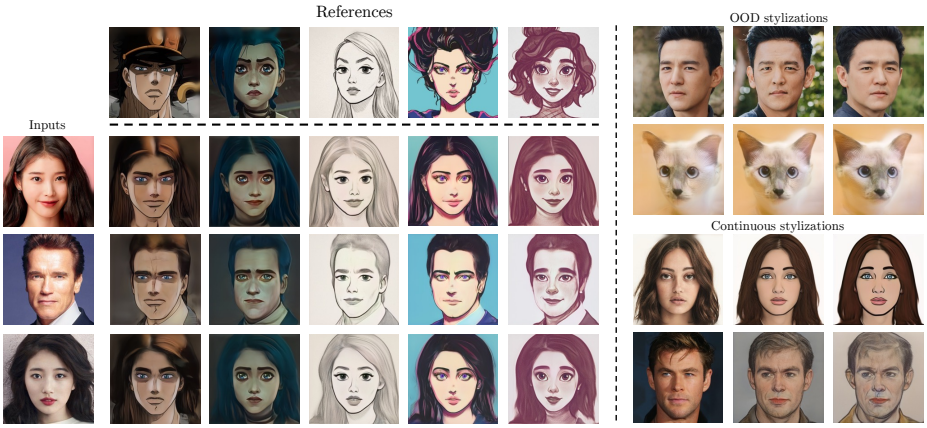

Fig. 1. JoJoGAN accepts a single style reference image (**top row**) and very quickly produces a style mapper that accepts an input (**left column**) and applies the style to that input. JoJoGAN can use extreme style references (**OOD stylizations**; the cat faces are JoJoGAN outputs for the human inputs above. Furthermore, JoJoGAN can apply styles to different extents (**Continuous stylization**); each row shows input; lightly stylized output; and strongly stylized output.

Abstract. A *style mapper* applies some fixed style to its input images (so, for example, taking faces to cartoons). This paper describes a simple procedure – JoJoGAN – to learn a style mapper from a single example of the style. JoJoGAN uses a GAN inversion procedure and StyleGAN's style-mixing property to produce a substantial paired dataset from a single example style. The paired dataset is then used to fine-tune a StyleGAN. An image can then be style mapped by GAN-inversion followed by the fine-tuned StyleGAN. JoJoGAN needs just one reference and as little as 30 seconds of training time. JoJoGAN can use extreme style references (say, animal faces) successfully. Furthermore, one can control what aspects of the style are used and how much of the style is applied. Qualitative and quantitative evaluation show that JoJoGAN produces high quality high resolution images that vastly outperform the current state-of-the-art.

Keywords: Generative Models, One-shot stylization, StyleGAN, Style Transfer

1 Introduction

A *style mapper* applies some fixed style to its input images (so, for example, taking faces to cartoons). This paper describes a simple procedure to learn a style mapper from a single example of the style. Our procedure allows, for example, an unsophisticated user to provide a style example, and then apply that style to their choice of image. Because stylizing face images – make me look like JoJo – is so desirable to unsophisticated users, we describe our method in the context of face images; but the method applies to anything.

To be useful, a procedure for learning a style mapper should: be easy to use; produce compelling and high quality results; require only one style reference, but accept and benefit from more; allow users to control how much style to transfer; and allow more sophisticated users to control what aspects of the style get transferred. We demonstrate with qualitative and quantitative evidence that our method meets these goals.

Learning a style mapper is hard, because the natural method – use paired or unpaired image translation [40,13,4] – isn't really practical. Collecting a new dataset per style is clumsy, and for many styles – Lucien Freud portraits, say – there may not be all that many examples. One might use few-shot learning techniques to fine-tune a StyleGAN [16] by adjusting the discriminator (as in [24,20,29,23]). But these methods do not have detailed supervision from pixel-level losses and so mostly fail to capture distinct style details and diversity.

In contrast, JoJoGAN (our procedure) takes a reference image (or images – but one image is enough) and makes a paired dataset using GAN inversion and StyleGAN's style-mixing property. This paired dataset is used to fine-tune StyleGAN using a novel direct pixel-level loss. The mechanics are straightforward: we can obtain a mapper (and so a rich supply of stylized portraits) from a single reference image in under a minute. JoJoGAN can use extreme style references (say, animal faces) successfully. Natural procedures control what aspects of the style are used and how much of the style is applied. Qualitative examples show that the resulting images look much better than alternative methods produce. Quantitative evidence strongly supports our method. Training and demo code is available at `https://github.com/mchong6/JoJoGAN`.

2 Related Work

Style transfer methods likely start with [7,10]; these are one shot methods, but do not result in style mappers in any natural way. Neural style transfer (NST) methods start with [9]; Johnson *et al.* offer a learned mapper, trained with a large dataset and Gatys *et al.*'s procedure to stylize [14]. In contrast, our method uses much less data and produces much higher resolution images. A rich literature

has followed, but general style transfer methods (for example [12,19,26]) cannot benefit from the detailed semantic and structural information captured by a GAN. Style transfer evaluation is mostly qualitative, but see [38]. Deformable Style Transfer (DST) [17] corrects structural errors by estimating spatial warps, then performing traditional neural style transfer; DST achieves impressive one-shot stylization, but warp estimation errors have significant effects and are hard to avoid (Figure 10).

StyleGAN [15,16] remains the state-of-the art unconditional generative model due to its unique style-based architecture. StyleGAN's AdaIN modulation layers (originally from [12]) have been shown to be disentangled and exhibit impressive editability [31,32,2]. StyleGAN has also been used as a prior for numerous tasks such as superresolution [22] and face restoration [35]. Pinkney *et al*. [27] first showed that finetuning the StyleGAN on a new dataset and performing layer swapping allows the StyleGAN to learn image to image translation with a relatively small dataset. But even obtaining a small paired dataset is hard: collection is difficult and expensive; one needs a new dataset for each new style; and in some cases (for example, Lucien Freud portrait style) there won't be many style images in the first place. In contrast, JoJoGAN creates a paired dataset from a single style reference by manipulating a pretrained StyleGAN2 [16] and a GAN inversion procedure, then finetunes using the created dataset.

One shot learning covers many applications (detection; classification; image synthesis), and methods remain specialized to their application. This paper focuses on one-shot image stylization, with a particular emphasis on faces.

One shot face stylization is now established. Learning a style mapper from very few examples results in overfitting problems. To control overfitting, [20,25] introduce regularization terms while [29,23] enforces constraints in the network's weights. These methods need tens to hundreds of style example images; in contrast, JoJoGAN works with one. Furthermore, these methods have difficulty capturing small style details, likely because they rely on an adversarial loss. BlendGAN [21] introduced a VGG-based style encoder and a weight blending module to learn arbitrary face stylization over a large styled faces dataset. As our comparisons show, this method fails to capture small but pertinent style details in face images. StyleGAN-NADA [8] uses CLIP [28] to perform zero/one shot image stylization based on text/image prompts, resulting in very strong generalization; as our comparisons show, StyleGAN-NADA fails to capture minute facial details that are important for face stylization.

Most similar to JoJoGAN is work by Zhu *et al*. [41] (detailed experimental comparison in Figure 11 and Appendix); this also uses uses GAN inversion to find a corresponding real face from a reference, so creating a paired datapoint. Zhu *et al*. use this simple datapoint and a number of CLIP-based losses (from [28]). In contrast, JoJoGAN creates a large dataset of paired datapoints from a single one, and so needs only a simple pixel loss (with an optional identity loss). Zhu *et al*. use gradient descent inversion II2S (from [42]), which is slow but more accurate. In contrast, JoJoGAN uses feed forward inversion based on a simple

encoder. Complex losses and slow inversion procedures mean Zhu *et al*. require some 15 minutes to train on a Titan XP; in contrast, JoJoGAN require 1.

3 Methodology

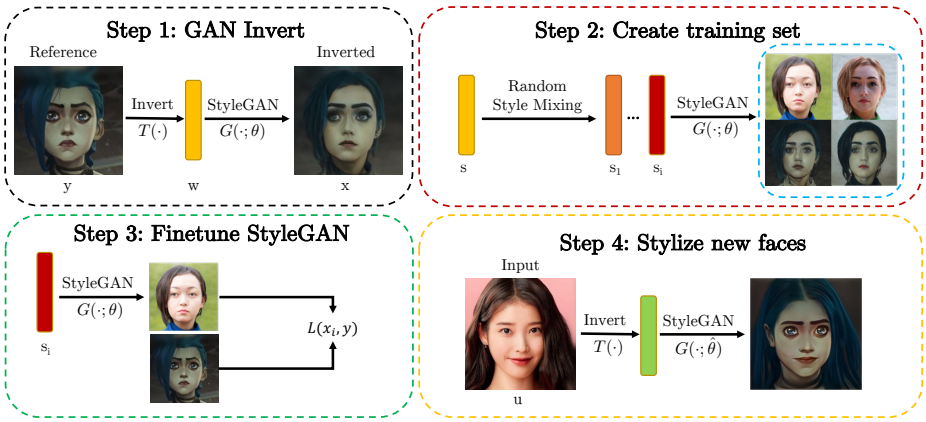

Fig. 2. Workflow: JoJoGAN's steps are: **GAN Inversion** to obtain a code s from the style reference; creating a **training set** $\mathcal{S}$ of similar s_i via random style mixing; **finetuning** a StyleGAN to obtain $\hat{\theta}$ so that $G(w_i;\hat{\theta}) \approx y$ using our perceptual loss; and **inference** by computing $G(T(u);\hat{\theta})$ for input u.

Write T for GAN inversion, G for StyleGAN, s for style parameters in StyleGAN's $\mathcal{S}$-space (notation after [36]; mixing in $\mathcal{S}$-space works better, see Appendix A.3), and θ for the parameters of the vanilla StyleGAN. JoJoGAN uses four steps (Figure 2):

1. **GAN inversion:** We GAN invert the reference style image y to obtain a style code $w = T(y)$ and from that a set of s parameters $s(w)$.
2. **Training set:** We use s to find a set of style codes $\mathcal{S}$ that are "close" to s. Pairs (s_i, y) for $s_i \in \mathcal{S}$ will be our paired training set.
3. **Finetuning:** We finetune the StyleGAN to obtain $\hat{\theta}$ such that $G(s_i;\hat{\theta}) \approx y$.
4. **Inference:** For input u, our stylized face is $G(s(T(u));\hat{\theta})$ (so $G \circ s \circ T$ is our style mapper).

Step 1: GAN Inversion: Remarkably, for any but extreme face style references y, we have $G(s(T(y));\theta)$ is a realistic – rather than stylized – face image (eg Figure 2, step 1). This is likely because a GAN inverter is trained to produce codes that result in realistic faces, does not see stylized faces in training, and so fails to generalize properly – in this context, a useful property.

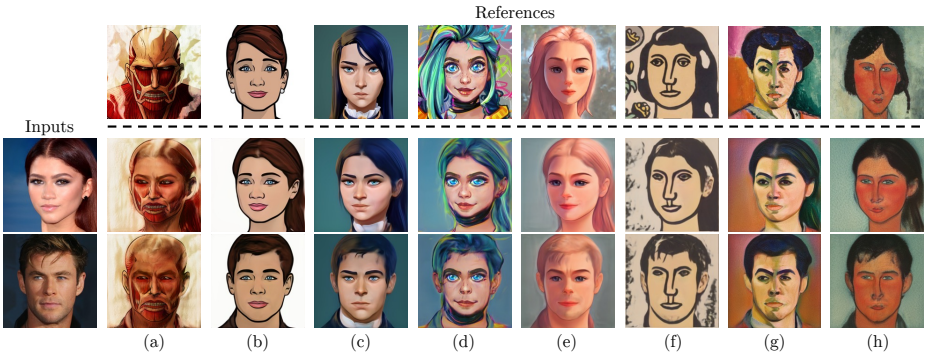

Fig. 3. JoJoGAN takes a single style reference image and produces a style mapper (reference images on top row; inputs far left). Note: clear following of input gender; subtle style details transferred (chin dimples in c; lip specularities in d); style lighting preserved (c, e); strong style effects in output, even from difficult styles (f, g, h); style idiosyncracies preserved (muscle fiber in a; bent nose in h; earrings in b).

Step 2: Training set: We must find a set of style codes $\mathcal{S}$ that are "close" to $s(w)$. We use StyleGAN's style mixing mechanic. We use a 1024 resolution StyleGAN2 with 26 style modulation layers, so $s \in \mathbb{R}^{26\times 512}$. Write $M \in \{0,1\}^{26}$ for a fixed mask, FC for the style mapping layer of the StyleGAN and $z_i \sim \mathcal{N}(0, I)$. We produce new style codes using

$$s_i = M \cdot s + (1 - M) \cdot s(FC(z_i)) \tag{1}$$

(and do so per batch). Different M result in different stylization effects (Section 4).

Step 3: Finetuning StyleGAN: We now assume that a properly trained style mapper will map $s_i \in \mathcal{S}$ to y. This assumption certainly works, and is reasonable when the style mapper "reduces information" – so, for example, mapping faces with slightly different eye sizes or hair textures to the same reference image. We finetune StyleGAN to obtain

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \, \text{loss}(\theta) = \underset{\theta}{\text{argmin}} \, \frac{1}{N} \sum_{i}^{N} \mathcal{L}(G(s_i; \theta), y) \tag{2}$$

where $\mathcal{L}$ is a novel perceptual loss (this choice is important; Section 3.1).

Step 4: Inference: For input u, our stylized face is $G(s(T(u)); \hat{\theta})$ (so $G \circ s \circ T$ is our style mapper). We could also generate random stylized samples by sampling random noise and generating with our finetuned StyleGAN.

3.1 Perceptual loss

The choice of loss in Equation 2 is important (Figure 4). While LPIPS [39] is a natural choice, it produces methods that lose detail. LPIPS is built on a VGG [33]

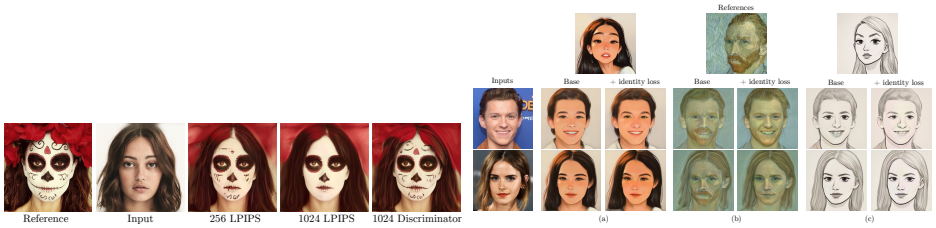

Fig. 4. Left: The choice of loss is important (this example is typical). For the style reference and the face shown, we train JoJoGAN using different losses. LPIPS at resolution 256 resolution leads to a loss of detail due to downsampling. LPIPS at 1024 does not control details, as the VGG filters (trained at 224) are not adapted to this scale. We match activations at layers of the pretrained discriminator from FFHQ-trained StyleGAN to compute a perceptual loss that preserves detail better. **Right:** some style inputs can result in outputs that lose identity (beard in b, for example). A straightforward identity loss can successfully control this effect, details in text.

backbone trained at a 224×224 resolution, but StyleGAN produces 1024×1024 images. The standard way to handle this mismatch is to downsample the images to 256×256 before computing LPIPS [16,34,1]. But this downsampling means we cannot control fine-grained details, which are mostly lost. Similarly, computing LPIPS at the native 1024 resolution leads to a complete loss of fine-grained detail as the VGG filters are not adapted to this resolution.

The pretrained StyleGAN discriminator is trained at the same resolution as the generator. The training process means that discriminator computes features that do not ignore details (otherwise the generator could produce low detail images). Discriminator features are known to stabilize GAN training when averaged over batches [30]. We choose to use the difference in discriminator activations at particular layers, per image (details in Appendix A.5). Write $D(\cdot)$ for the activations; then $\mathcal{L}(G(s_i;\theta), y) = \parallel D(G(s_i;\theta)) - D(y) \parallel_1$. A version of this loss is used in GPEN [37] but to our knowledge, we are the first to compare it with others and show how effective it is.

4 Variants

Controlling Identity: Some style references distort the original identity of the inputs (Figure 4). In such cases, writing sim for cosine similarity and F for a pretrained face embedding network (we use ArcFace [6]), we use

$$\mathcal{L}_{id} = 1 - sim(F(G(s_i;\theta)), F(G(s_i;\hat{\theta}))) \tag{3}$$

to compel the finetuned network to preserve identity; we use it only for references that severely distort the identity and note in captions when we use it (eg. Figure 4(b)).

Controlling Style Intensity by Feature Interpolation: Feature interpolation [5] allows us to vary the intensity of the style. Let f_i^A be the layer i

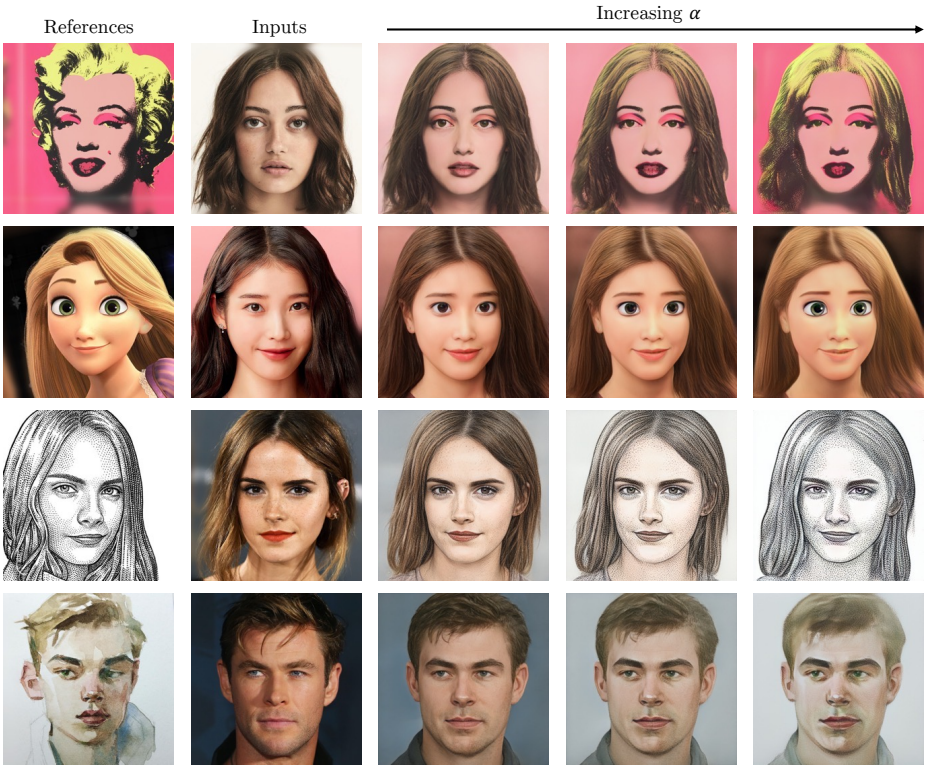

Fig. 5. Feature interpolation allows a user to control style intensity. As α increases, the results take the style of the reference more strongly.

intermediate feature maps from the original StyleGAN and f_i^B from JoJoGAN; then we can perform continuous face stylization by using $f = (1-\alpha)f_i^A + \alpha f_i^B$ where α is the interpolation factor. Increasing α results in stronger style intensity (Figure 5).

Extreme Style References: For JoJoGAN to work, $\mathcal{S}$ has to consist of s_i that produce sensible responses from the StyleGAN. If the style reference is (roughly) a human face, there are no problems. An extreme style reference image is one where GAN inversion produces s that is out of distribution for the StyleGAN, for example, an image of an animal face. We are not aware of any test (other than trying) to distinguish between extreme and standard style references, but Figure 19 in the Appendix demonstrates that using s from GAN inversion on animal faces results in poor style transfer. For extreme style references y, rather than use $s(T(y))$ to construct $\mathcal{S}$, we use the mean style code $\overline{s} = \sum_1^{10000} s(FC(z \sim \mathcal{N}(0, I)))$ (note this style code is the best possible estimate of $s(T(y))$ for an image y *that one does not have*). With this modification, JoJoGAN works well on extreme style references (Figure 6; note how the animal head poses are controlled by the input images).

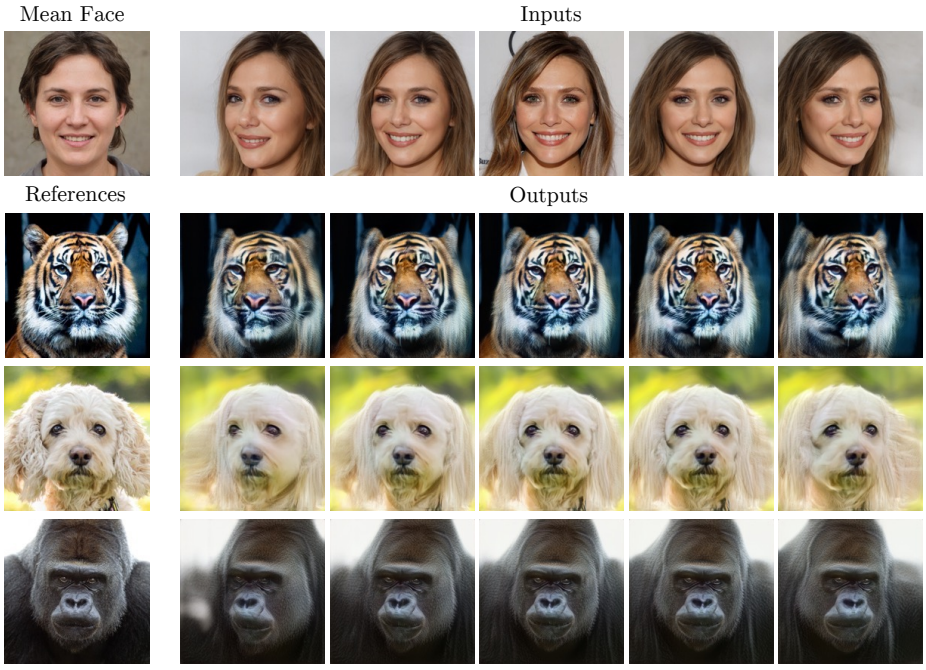

Fig. 6. OOD references and using $\overline{w}$: JoJoGAN is able to handle OOD references that do not invert well by using mean style code $\overline{w}$. Even on animal faces which are semantically very different the human faces StyleGAN was trained on, JoJoGAN can generate realistic animal faces with poses that matches the input.

Multi-shot Stylization: JoJoGAN extends to multi-shot stylization in the natural way (use each reference to construct a $\mathcal{S}_k$ for each reference y_k; now finetune using

$$\frac{1}{M * N} \sum_{j}^{M} \sum_{i}^{N} \mathcal{L}(G(s_{ij}; \theta), y_j).$$

Using more than one reference produces small but useful qualitative improvements in the style mapper (Figure 12)

5 Controlling Aspects of Style

Style transfer is intrinsically ambiguous. The output should be "like" the reference as to style, and "like" the input as to content, but the distinction between content and style is vague. JoJoGAN offers methods to choose whether (say) the output should have exaggerated eyes (like the reference) or more natural eyes (like the input). Simple control is obtained by choice of mask and by loss. More detailed control follows by careful attention to the GAN inversion.

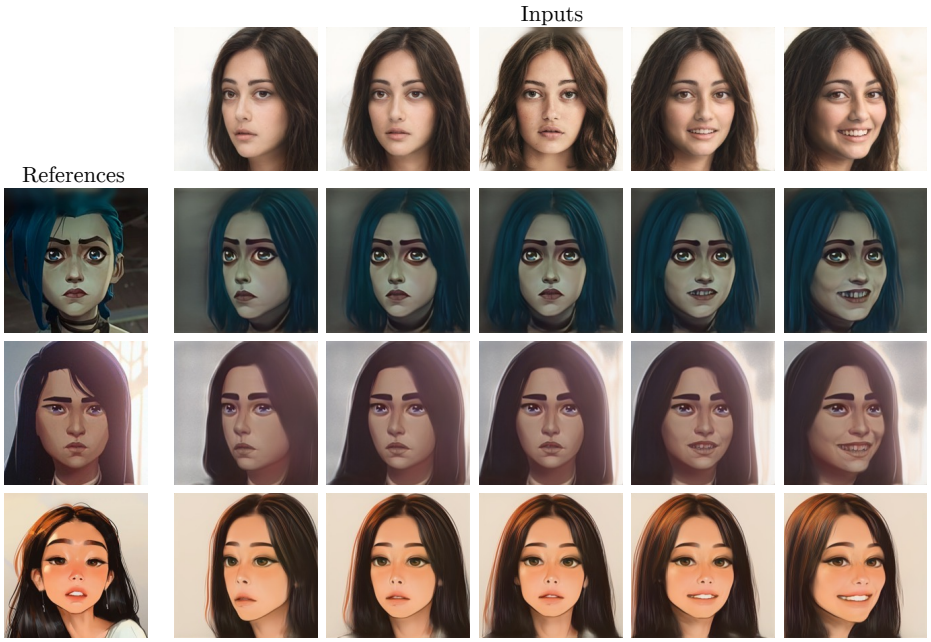

Fig. 7. JoJoGAN produces smooth and consistent stylization as the face moves and changes expressions.

Controlling Aspects of Style by Mask Choice and by Loss: Different choices of M will produce significant differences in $\mathcal{S}$, and so in results. Replacing too many elements of s with random numbers may result in a JoJoGAN that maps every face to the style reference; replacing too few means finetuning sees too few examples. Furthermore, replacing elements at locations corresponding to different StyleGAN layers controls different effects (see [16]). Figure 8 demonstrates this choice has significant effects by displaying results from two different M. The first gives dataset $\mathcal{X}$, the second $\mathcal{C}$. Both masks are chosen to maintain the input face pose and hairstyles while allowing features such as eye sizes and textures to vary, so the mask has ones in locations known to correspond to pose and zeros in those known to correspond to eye-sizes, see [2]. But $\mathcal{C}$ is chosen so that the color of the input is preserved (so ones in relevant locations); and $\mathcal{X}$ so that color is driven by the style example. To ensure that the color of the input is preserved for the $\mathcal{C}$ case, we apply the loss in Equation (2) to grayscale versions of the relevant images. This means the StyleGAN is finetuned to obtain the spatial appearance of the style target, but not its colors (variants in Appendix A.4)

5.1 Control by Manipulating GAN Inversion

The choice of GAN inverter matters. If the GAN inverter produces an extremely realistic face from the reference, JoJoGAN will be trained to map s_i that rep-

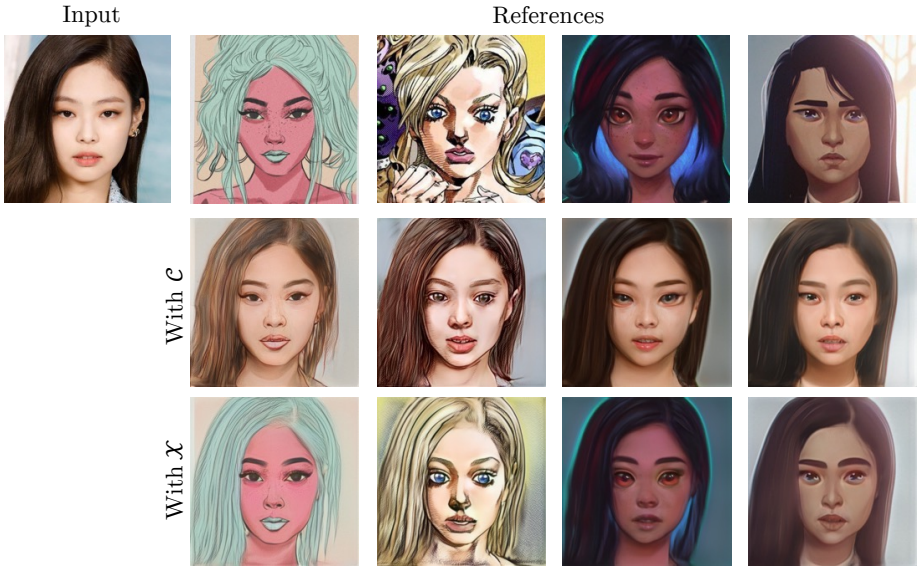

Fig. 8. The aspects of style that are transferred can be chosen using M. The text describes our procedures to create two different datasets $\mathcal{C}$ and $\mathcal{X}$ from different choices of M that yield different stylizations. Finetuning using $\mathcal{C}$ mostly preserves the colors of the input; finetuning using $\mathcal{X}$ mostly reproduces the colors of the style example.

resent highly realistic faces to the style reference, and so will tend to produce aggressively stylized faces. By the same argument, if the GAN inverter produces a somewhat stylized face from the reference, JoJoGAN will tend to produce lightly stylized faces and to preserve the features of the input face (so an input with small eyes will result in an output with small eyes, say – example in Appendix Figure 14). This effect can be used to control how much and what style is transferred by blending inverted codes.

Using two GAN inverters is clumsy in practice, but recall the mean style code is the best possible estimate of $s(T(y))$ for an image y *that one does not have*), and so is the output of a (rather bad, but very fast) GAN inverter. We produce a virtual inverter $V(y)$ by blending the code produced by our standard inverter with the mean, using the procedure of Section 3 (but a different mask M). The blend is adjusted so that $G(s(V(y)); \theta)$ has desirable properties (so, for example, to preserve the eyes of the reference, $G(s(V(y)); \theta)$ should have realistic eyes). We then apply the JoJoGAN pipeline using V rather than T to generate training data. Using V rather than T in training changes the pairs (s_i, y) used in finetuning, and so the behavior of G. At inference, we compute $G(s(T(u); \hat{\theta})$ as before. Figure 9 demonstrates the extent of our style control. In Figure 9(b), using the blended inversion gives us larger eyes and thicker lips compared to using the accurate inversion (a). Further detail on blending the inverter in Appendix A.1.

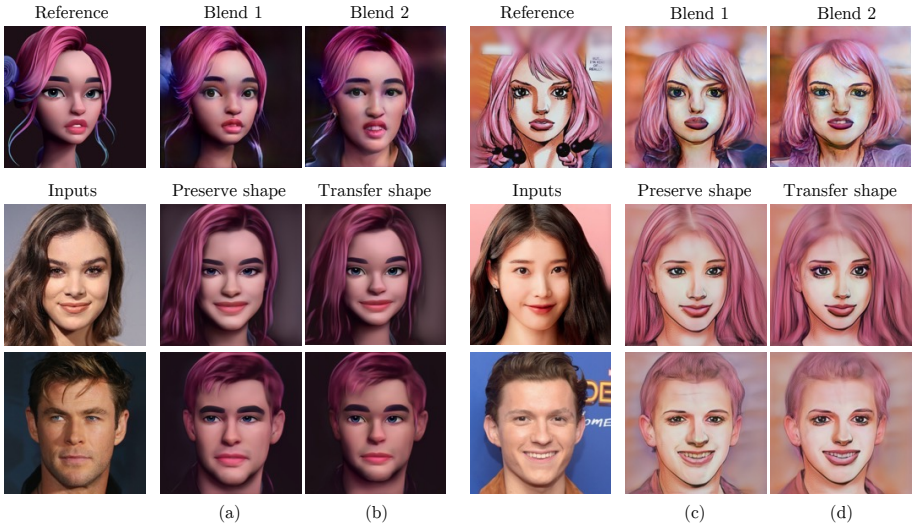

Fig. 9. How style is transferred can be controlled by blending the codes from two GAN inverters, then applying the JoJoGAN pipeline. (for these examples, one inverter just produces the mean code). For each reference, **top** shows $G(s(V(y)); \theta)$ for different blends. Notice how blending the inverter codes produces substantial changes in the inversion (eg **left** reference). By choice of blend, we can produce style mappers that tend to **preserve** the shape of input eye, nose and face or to **transfer** shapes from the reference. So, for example, (a) and (c) have eyes more like the input; but (b) and (d) have larger eyes, more like the reference. (b) has significantly smaller faces than (a).

6 Experiments

Setup: For GAN inversion, we use ReStyle [1]. We finetune JoJoGAN for 200 to 500 iterations depending on the reference with Adam optimizer [18] at a learning rate of 2×10^{-3}. Finetuning on an Nvidia A40 takes about 30 to 60 seconds.

Qualitative evaluation: A style mapper should: produce good looking outputs; faithfully transfer features from the style reference; and preserve the identity of the input. Qualitative evaluation shows JoJoGAN has these properties and vastly outperforms current methods.

Comparisons: Figure 10 shows comparisons of JoJoGAN to the state-of-the-art one/few shot stylization methods StyleGAN-NADA [8], BlendGAN [21], Ojha *et al*. [24] and DST [17]. JoJoGAN captures small details well that define the style while maintaining the identity of the input face well. JoJoGAN results are typically improved when there are multiple consistent style references. Figure 12 compares several one-shot stylizations of each of a set of examples with a multi-shot stylization using all. Notice that one-shot stylization copies effects from the style reference aggressively (as it must), whereas when there are multiple style examples, JoJoGAN is able to blend details to hew more closely to the input.

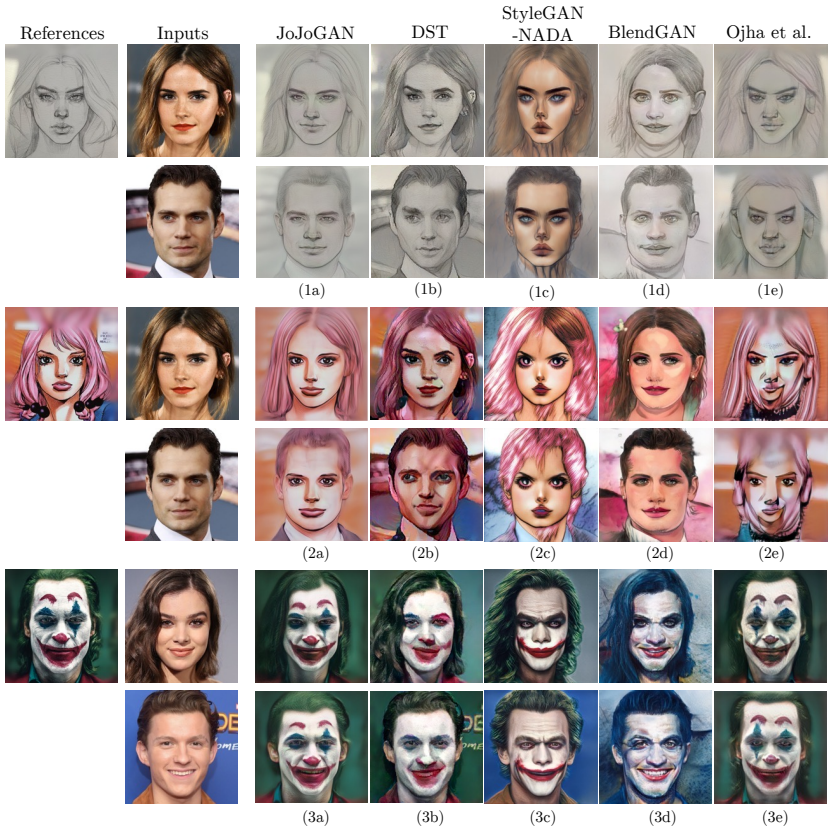

Fig. 10. JoJoGAN offers visible qualitative improvements over current SOTA methods for one shot face stylization. JoJoGAN captures the distinctive rendering style of the reference while preserving input pose, expression and identity. Note: excessive contrast (1b); color errors (1c, 2b, 3d); distorted facial layout (d, e); chin shape (b).

Figure 11 shows a comparison with [41] (two examples in figure; others – except 2, for which we cannot find source – in supplementary). Note we can use only references shown in their paper, as the method is not open sourced.

Quantitative Evaluation by User study: We proceed in two stages, to reduce choice fatigue for users. From Figure 10, DST gives good results in most cases while other methods produces examples with severe problems. We therefore compare JoJoGAN to non-DST methods in a first study, and to DST in a second. In each, users see a style reference, an input face, and stylizations from the methods and are asked to choose the stylization that best captures the style reference and while preserving the original identity. The first study resulted in a total of 186 responses from 31 participants who overwhelmingly prefer JoJoGAN to other methods at 80.6%; the effect is so large that no significance issues

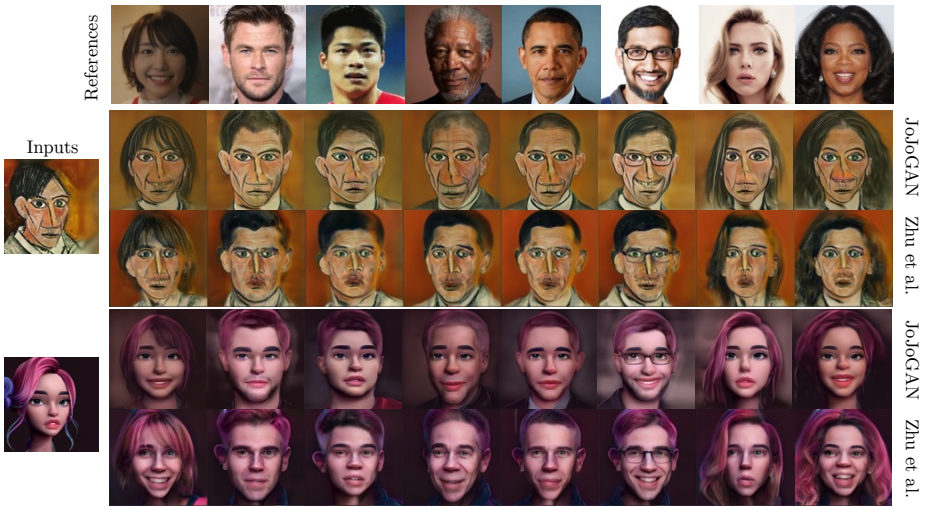

Fig. 11. We compare with Zhu *et al*. [41] on two examples for references used in their paper and described as hard cases there (others in supplementary). For each reference, the top row is JoJoGAN while the second row is Zhu *et al*. Note how their method distorts chin shape, while JoJoGAN produces strong outputs.

arise. The second study gathered 96 responses from 16 participants who prefer JoJoGAN to DST at 74%.

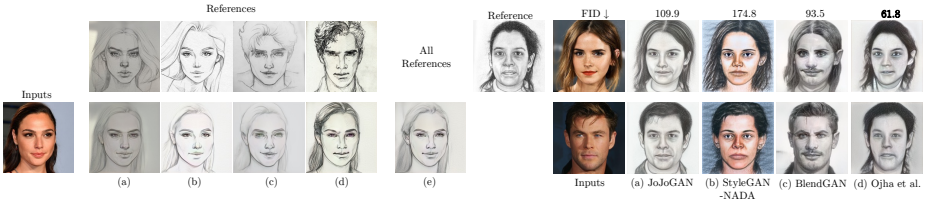

Fig. 12. Left: JoJoGAN's method extends cleanly to deal with multiple style references, if they are available. The figure compares one-shot stylizations of a reference with a multi-shot stylization for one input (more in supplementary). Note aggressive copying in the case of a single reference, including: noses in (a); lips in (b) and (c); and chin dimples in (b) and (d). This effect is notably muted when more references are available (and JoJoGAN can blend details from references), so (e) mouth and chin follow the input more closely. **Right:** JoJoGAN's FID score on the sketches dataset [24] is significantly larger than that of the best comparison. BlendGAN gets a better FID, but does not capture reference style well (note strong shading gradients, absent from the style reference); Ojha *et al*. get the best FID, but impose strong distortions on the input face (note other comparisons in Figure 10).

Quantitative Evaluation by FID: FID [11] is a metric that is widely used to evaluate the quality and diversity of generated images by comparing population statistics. FID can be used to evaluate style mappers as follows [25]. Randomly select a reference from the style dataset and performing one shot stylization with it; now stylize a set of face images and compute the FID between the result and the original style dataset. To compute FID, we perform one shot stylization using the sketches dataset [24] and compute FID using the test set. JoJoGAN scores well behind SOTA on this metric. We report FID for JoJoGAN for candor and show FID for SOTA comparisons in Figure 12, but point out that FID is a poor metric for style mappers. The procedure described cannot measure the fidelity with which the mapper preserves the input (for example, the FID for the completely ineffectual mapper that just produces a random sample from the style dataset would be close to zero). Further, a perfect style mapper might produce a high FID with the protocol described, because its stylized images should be biased toward the input (for example, a perfect mapper with only male input images should produce a population of sketches that is not close to the original set of sketches). Finally, the datasets used for stylization are often very small (290 in the case of the sketches dataset), and computing FID for a small dataset is dangerous due to large biases [3].

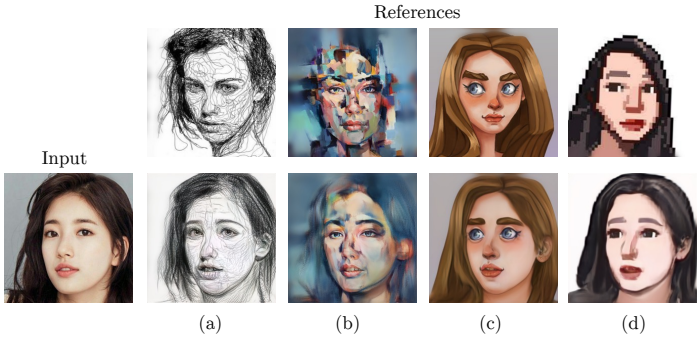

Fig. 13. Some style references are hard for JoJoGAN, likely a result of complicated structures in the style reference that are unfamiliar to StyleGAN. Note: loops in (a) mapped to strokes in the output; structure of brush strokes in (b) being broken up in output; gaze direction in (c) controlled by style reference rather than by input; high frequency pixel grids in (d) map to smooth strokes.

Failures: Using too small a $\mathcal{S}$ leads to problems (Appendix Figure 17), typically artifacts and missing style details. As JoJoGAN only sees a single style reference, it does not always work for all style references. One common issue JoJoGAN has is that the eye gaze direction is often driven by the reference image rather than the input. The intended behavior is to preserve the gaze direction of the original input, yet JoJoGAN copies the reference instead. Figure 13 shows results on very difficult references, illustrating visual failure modes.

References

1. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2021)
2. Chong, M.J., Chu, W.S., Kumar, A., Forsyth, D.: Retrieve in style: Unsupervised facial feature transfer and retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3887–3896 (October 2021)
3. Chong, M.J., Forsyth, D.: Effectively unbiased fid and inception score and where to find them. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6070–6079 (2020)
4. Chong, M.J., Forsyth, D.: Gans n' roses: Stable, controllable, diverse image to image translation (works for videos too!) (2021)
5. Chong, M.J., Lee, H.Y., Forsyth, D.: Stylegan of all trades: Image manipulation with only pretrained stylegan. arXiv preprint arXiv:2111.01619 (2021)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
7. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. Proceedings of SIGGRAPH 2001 pp. 341–346 (August 2001)
8. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021)
9. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
10. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. Proceedings of SIGGRAPH 2001 pp. 327–340 (August 2001)
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
12. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
13. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018)
14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (2016)
15. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
16. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proc. CVPR (2020)
17. Kim, S.S.Y., Kolkin, N., Salavon, J., Shakhnarovich, G.: Deformable style transfer. In: European Conference on Computer Vision (ECCV) (2020)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2015)
19. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Advances in Neural Information Processing Systems (2017)

20. Li, Y., Zhang, R., Lu, J.C., Shechtman, E.: Few-shot image generation with elastic weight consolidation. In: Advances in Neural Information Processing Systems (2020)
21. Liu, M., Li, Q., Qin, Z., Zhang, G., Wan, P., Zheng, W.: Blendgan: Implicitly gan blending for arbitrary stylized face generation. In: Advances in Neural Information Processing Systems (2021)
22. Menon, S., Damian, A., Hu, S., Ravi, N., Rudin, C.: Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 2437–2445 (2020)
23. Mo, S., Cho, M., Shin, J.: Freeze the discriminator: a simple baseline for fine-tuning gans. In: CVPR AI for Content Creation Workshop (2020)
24. Ojha, U., Li, Y., Lu, C., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: Few-shot image generation via cross-domain correspondence. In: CVPR (2021)
25. Ojha, U., Li, Y., Lu, J., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: Few-shot image generation via cross-domain correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10743–10752 (2021)
26. Park, D.Y., Lee, K.H.: Arbitrary style transfer with style-attentional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5880–5888 (2019)
27. Pinkney, J.N., Adler, D.: Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
29. Robb, E., Chu, W.S., Kumar, A., Huang, J.B.: Few-shot adaptation of generative adversarial networks. arXiv preprint arXiv:2010.11943 (2020)
30. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29**, 2234–2242 (2016)
31. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE transactions on pattern analysis and machine intelligence (2020)
32. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1532–1540 (2021)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
34. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. arXiv preprint arXiv:2102.02766 (2021)
35. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9168–9178 (2021)
36. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)
37. Yang, T., Ren, P., Xie, X., Zhang, L.: Gan prior embedded network for blind face restoration in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

38. Yeh, M.C., Tang, S., Bhattad, A., Zou, C., Forsyth, D.: Improving style transfer with calibrated metrics. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
39. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
40. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Computer Vision (ICCV), 2017 IEEE International Conference on (2017)
41. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Mind the gap: Domain gap control for single shot domain adaptation for generative adversarial networks. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=vqGi8Kp0wM
42. Zhu, P., Abdal, R., Qin, Y., Femiani, J., Wonka, P.: Improved stylegan embedding: Where are the good latents? arXiv preprint arXiv:2012.09036 (2020)

A Appendix

A.1 Choice of GAN inversion

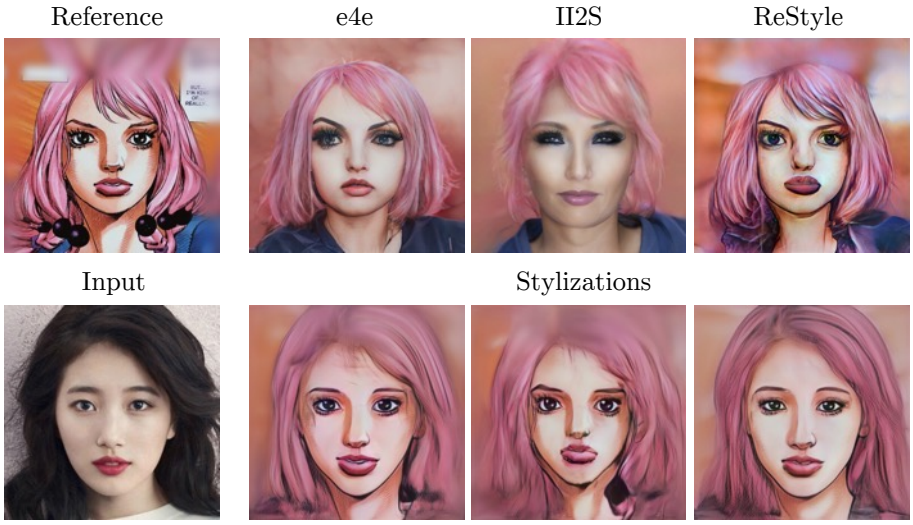

Fig. 14. The choice of GAN inversion matters. We compare JoJoGAN trained on e4e [34], II2S [42], and ReStyle [1] inversions. II2S gives the most realistic inversions leading to stylizations that preserves shapes and proportions of the reference. ReStyle gives the most accurate reconstruction leading to stylization that better preserves the features and proportions of the input.

JoJoGAN relies on GAN inversion to create a paired dataset. We investigate the effect of using 3 different GAN inversion methods, e4e [34], II2S [42], and ReStyle [1] in Figure 14.

Using e4e fails to accurately recreate the style reference and conveniently gives us a corresponding real face. On the other hand, ReStyle more accurately inverts the reference, giving a non-realistic face. II2S is a gradient-descent based method with a regularization term that allows us to map the style code to higher density region in the latent space. The regularization term results in a very realistic face that are somewhat inaccurate to the reference.

The different inversions give us different JoJoGAN results. Training with ReStyle leads to clean stylization that accurately preserves the features and proportions of the input face. Training with II2S on the other hand, leads to heavy stylization that borrows the shapes and proportions from the reference. However, this also leads to pretty heavy semantic changes from the input face and artifacts (note the change of identity, artifacts along the neck).

In practice, we blend the styles codes from ReStyle and the mean face. For M, we borrow the style code from mean face at layers 7, 9 and 11. This borrows

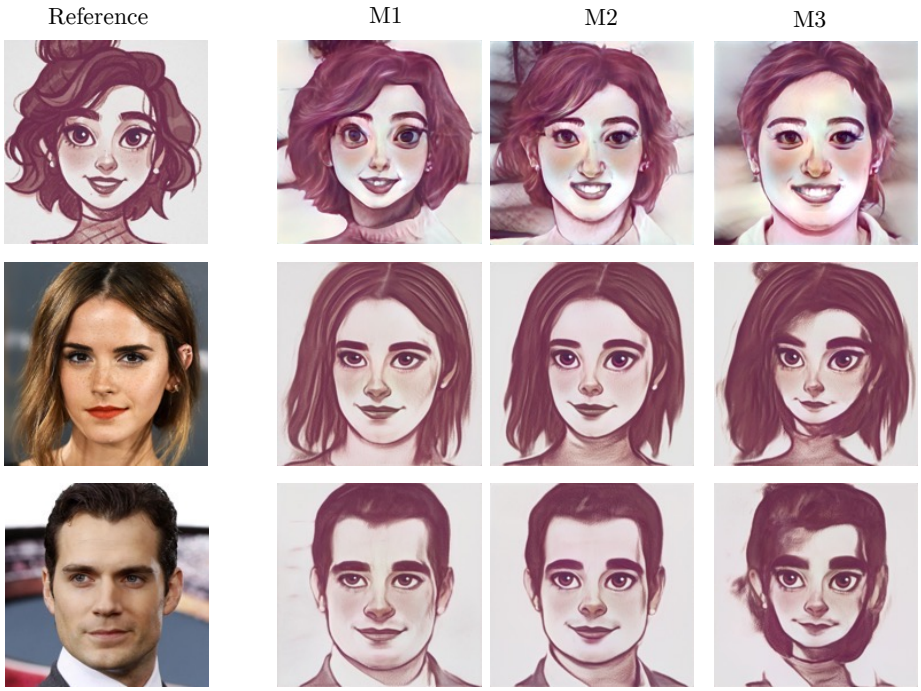

Fig. 15. The choice of M matters. M controls the blend between the inverted style with the mean style. $M1$ is the closest to the reference, leading to smaller features (e.g., eyes). $M3$ is the closest to a real face, leading to exaggerated features more like reference and also significant artifacts.

the facial features of the mean face to the inversion. However, it is impossible to only affect the proportions of the features by simply blending coarsely at a layer level. For example, naively blending the mean face can change the expression of the inversion, e.g. from neutral to smiling or introduce artifacts. We thus have to blend at a finer scale, which we are able to do so by isolating specific facial features in the style space using RIS [2]. Figure 15 compares the results of using different M for blending. Note that when the blended image is more face-like ($M3$), the exaggerated features of the reference is transferred. However, significant artifacts are introduced, see $M3$ row 2. By carefully selecting M, we can transfer the exaggerated features while avoiding artifacts, see $M2$.

A.2 Identity loss

Before computing identity loss, we grayscale the input images to prevent the identity loss from affecting the colors. The weight of the identity loss is reference dependent, but we typically choose between 2×10^3 to 5×10^3.

A.3 Choice of style mixing space

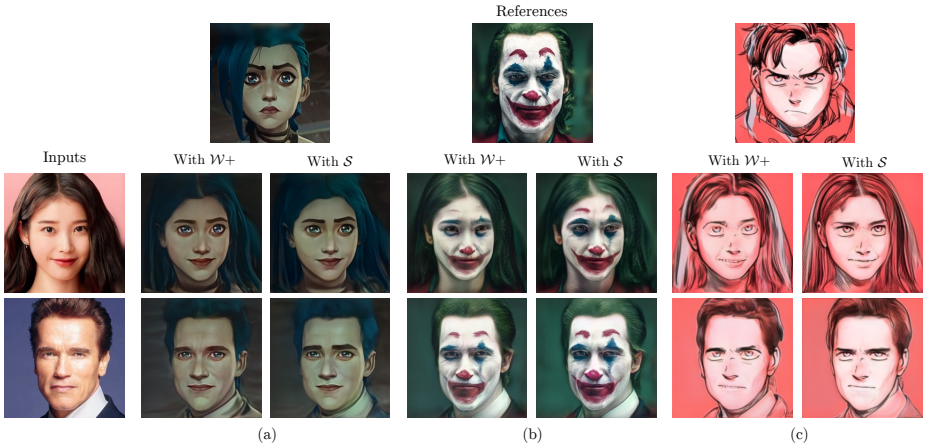

Fig. 16. We study how the choice of latent space to do style mixing affects JoJoGAN. Style mixing in $\mathcal{S}$ space gives more accurate color reproduction in (a) and (b) and better stylization effect (note the eyes) in (c).

Style mixing in Equation (1) allows us to generate more paired datapoints. It is reasonable to map faces with slight difference in textures, colors, to the same reference. As such it is pertinent that while we style mix to generate different faces, we need certain features such as identity, face pose, etc to remain the same. We study how the choice of latent space to do style mixing affects the stylization. In Figure 16 we see that style mixing in $\mathcal{S}$ gives better color reproduction and overall stylization effect. This is because $\mathcal{S}$ is more disentangled [36] and allows us to more aggressively style mix without changing the features we want intact.

A.4 Varying dataset

Using $\mathcal{C}$ and $\mathcal{X}$ gives different stylization effects. Finetuning with $\mathcal{X}$ accurately reproduces the color profile of the reference while $\mathcal{C}$ tries to preserve the input color profile. However this is insufficient to fully preserve the colors as we see in Figure 17. Grayscaling the images before computing the loss in Equation (2) in addition to finetuning with $\mathcal{C}$ gives us stylization effects without altering the color profile. We show that it is necessary to use both $\mathcal{C}$ and grayscaling to achieve this effect and using $\mathcal{X}$ and grayscaling is insufficient.

A.5 Feature matching loss

For discriminator feature matching loss, we compute the intermediate activations after resblock $2, 4, 5, 6$.

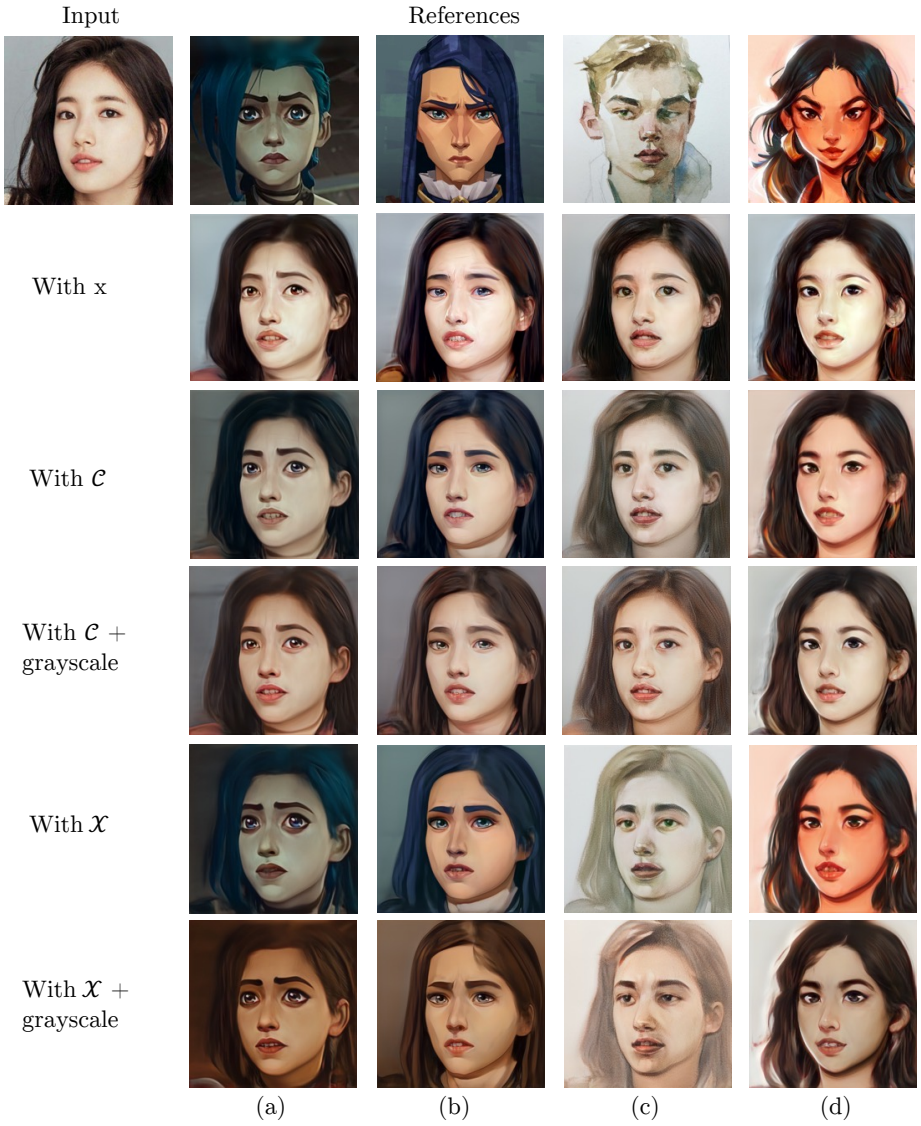

Fig. 17. The choice of training data has an effect. **First row:** when there is just one example in $\mathcal{W}$, JoJoGAN transfers relatively little style, likely because it is trained to map "few" images to the stylized example. **Second row:** same training procedure as in Figure 8, using $\mathcal{C}$. **Third row:** same training procedure as second row but with grayscale images for Equation (2). **Fourth row:** same training procedure as in Figure 8, using $\mathcal{X}$. **Fifth row:** same training procedure as Fourth row but with grayscale images for Equation (2).

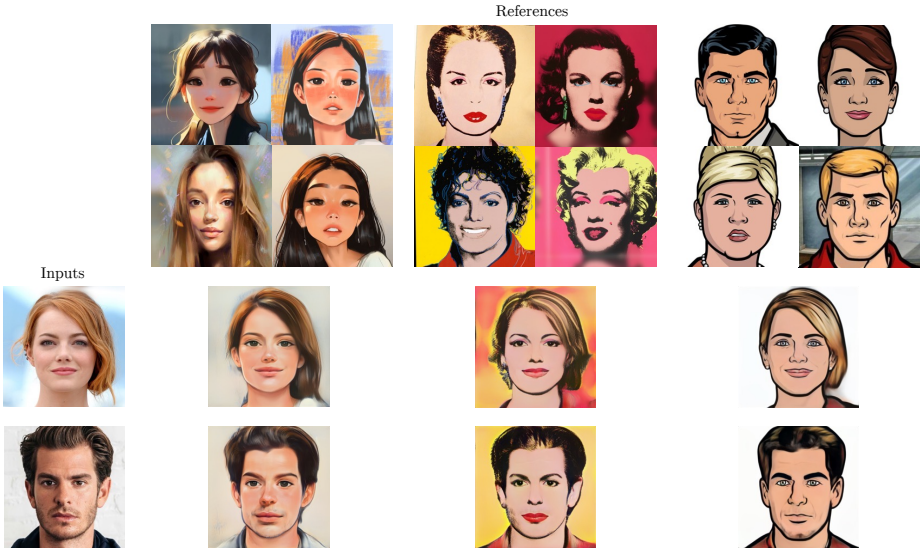

Fig. 18. More multi-shot examples

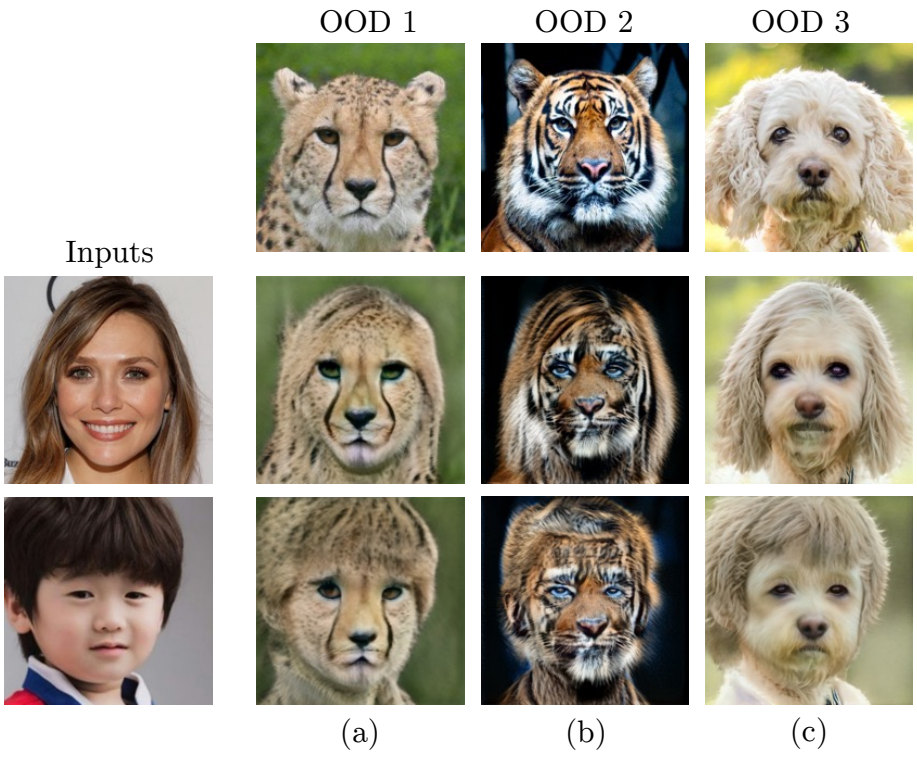

Fig. 19. JoJoGAN produces unsatisfactory style transfers on OOD cases, producing human-animal hybrids.

Fig. 20. We compare with Zhu *et al*. [41] on all examples for references used in their paper and described as hard cases there. For each reference, the top row is JoJoGAN while the second row is Zhu *et al*. Note how their method distorts chin shape, while JoJoGAN produces strong outputs.

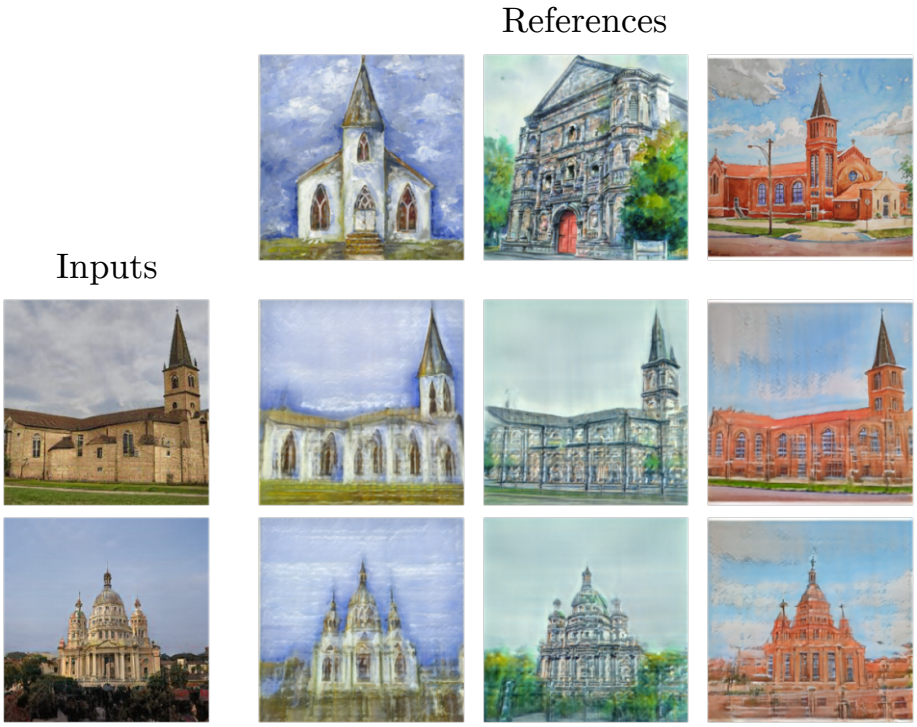

Fig. 21. JoJoGAN is a method to benefit from what a StyleGAN knows, and so should apply to other domains where a well-trained StyleGAN is available. Here we demonstrate JoJoGAN applied to LSUN-Churches.