HRInversion: High-Resolution GAN Inversion for Cross-Domain Image Synthesis

Peng Zhou, Lingxi Xie, Bingbing Ni, Lin Liu, and Qi Tian, *Fellow, IEEE*

***Abstract*—We investigate GAN inversion problems of using pre-trained GANs to reconstruct real images. Recent methods for such problems typically employ a VGG perceptual loss to measure the difference between images. While the perceptual loss has achieved remarkable success in various computer vision tasks, it may cause unpleasant artifacts and is sensitive to changes in input scale. This paper delivers an important message that algorithm details are crucial for achieving satisfying performance. In particular, we propose two important but undervalued design principles: (i) not down-sampling the input of the perceptual loss to avoid high-frequency artifacts; and (ii) calculating the perceptual loss using convolutional features which are robust to scale. Integrating these designs derives the proposed framework, HRInversion, that achieves superior performance in reconstructing image details. We validate the effectiveness of HRInversion on a cross-domain image synthesis task and propose a post-processing approach named local style optimization (LSO) to synthesize clean and controllable stylized images. For the evaluation of the cross-domain images, we introduce a metric named ID retrieval which captures the similarity of face identities of stylized images to content images. We also test HRInversion on non-square images. Equipped with implicit neural representation, HRInversion applies to ultra-high resolution images with more than 10 million pixels. Furthermore, we show applications of style transfer and 3D-aware GAN inversion, paving the way for extending the application range of HRInversion.**

***Index Terms*—GAN inversion, perceptual loss, image synthesis.**

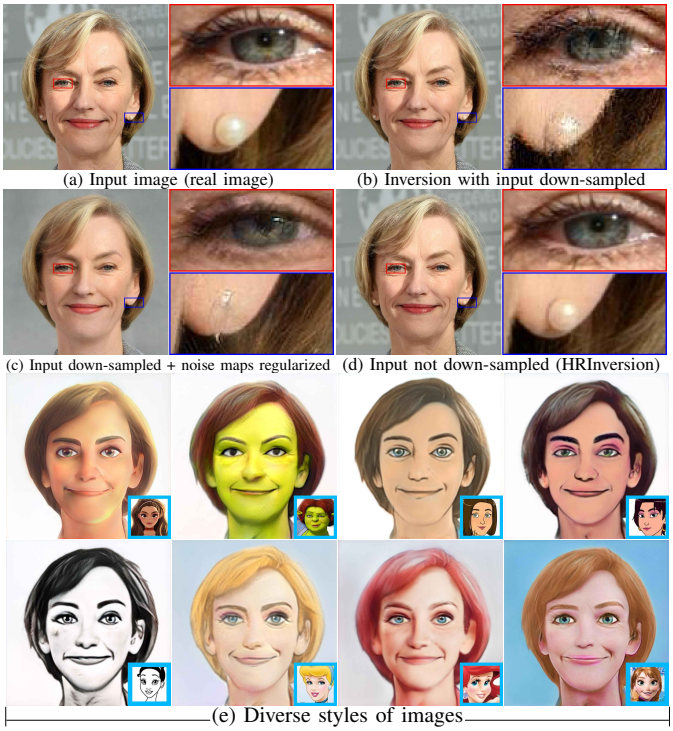


Fig. 1. (a) Input image. (b) Inverted image with input down-sampled for the perceptual loss. Down-sampling the input of the perceptual loss causes artifacts when optimizing the latent code $\boldsymbol{w}$ and the noise maps $\boldsymbol{n}$ simultaneously. (c) Inverted image with input down-sampled and noise maps regularized. Regularizing noise maps alleviates the artifacts but compromises the reconstruction quality. (d) Inverted image with input not down-sampled. The reconstructed image is close to the original input image. (e) Our method is capable of synthesizing various styles of cross-domain images without unpleasant noise. Please zoom in to see details.

I. Introduction

GENERATIVE Adversarial Networks (GANs) [1]–[10] have made considerable progress in generating photo-realistic images. Recently, there has been a growing interest in projecting real images into the latent space of pre-trained GANs, also known as GAN inversion [11]–[13]. Because the pre-trained generator contains prior knowledge, which is helpful for image restoration and editing, leveraging the prior has promoted a large number of tasks such as super-resolution and image manipulation [14]–[17].

To solve the GAN inversion task, a VGG perceptual loss has become a de-facto standard loss. The perceptual loss captures differences between high-level image feature representations extracted from a pre-trained VGG model. Compared to pixel-wise losses such as MSE, perceptual losses measure the semantic similarity of images rather than low-level pixel differences. It has been found that perceptual metrics are in line with human perception [18]. As such, perceptual losses have been applied to many tasks such as style transfer [19], [20], super-resolution [21], [22], and image-to-image translation [23], [24].

Although the VGG perceptual loss has many applications in computer vision, we find that people usually use it as an off-the-shelf module. There are a few comprehensive studies for perceptual losses [18], [21]. Nevertheless, previous works mainly focus on tasks such as super-resolution and style transfer. Some of their conclusions for perceptual losses may be out of date and no longer apply to the emerging GAN

This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also partially supported by Grant YG2021ZD18 from Shanghai Jiaotong University Medical Engineering Cross Research (Corresponding author: Bingbing Ni). Peng Zhou and Bingbing Ni are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: {zhoupengcv, nibingbing}@sjtu.edu.cn).

Lingxi Xie and Qi Tian are with Huawei Cloud BU, Shenzhen, Guangdong 518129, China (e-mail: 198808xc@gmail.com; tian.qi1@huawei.com). Lin Liu is with the University of Science and Technology of China, Hefei, Anhui 230052, China (e-mail: ll0825@mail.ustc.edu.cn).

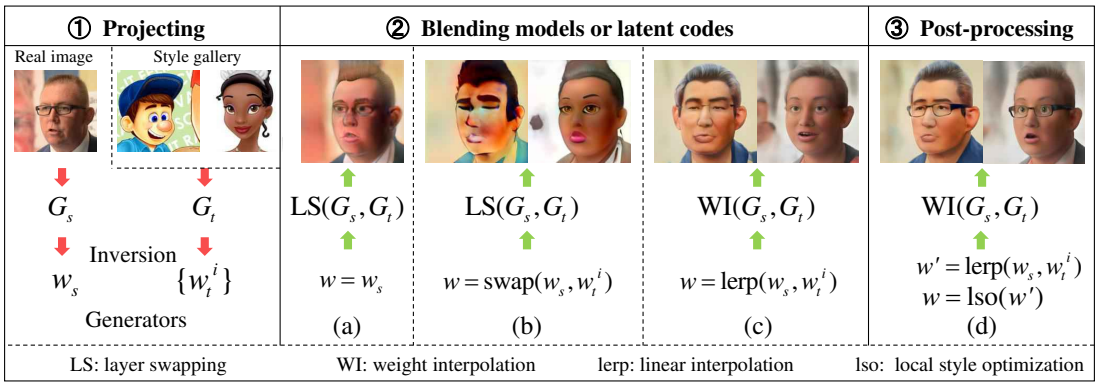


Fig. 2. The framework of cross-domain image synthesis. The common inversion-based cross-domain image synthesis method consists of two steps: (i) projecting the images into the latent space; and (ii) mixing latent codes or models. We propose a post-processing algorithm, local style optimization (LSO), as step 3, to further improve the synthesis quality. There are three mixing methods: (a) Only replace the high-resolution layers of the source generator with the corresponding layers of the target generator, *i.e.*, layer swapping [28]. The disadvantage of this method is that it only generates one style for a real image. (b) Replace not only the layers but also the latent code $\boldsymbol{w}$. We construct a style gallery $\{\boldsymbol{w}_t^i\}$ through GAN inversion. This method can generate images in a diverse style, but is not robust and may cause unpleasant noise. (c) Linearly interpolate the weights of the generators and the latent code $\boldsymbol{w}$. This method can fully compound image styles but at the cost of losing pose and semantic properties. (d) We propose to amend the interpolated latent code $\boldsymbol{w}'$ with a post-processing step, so as to restore pose and semantic properties of the real image and also remove noise (please see Sec. III-C for details).

inversion task. As a result, people may use perceptual losses inappropriately, leading to suboptimal results for their tasks.

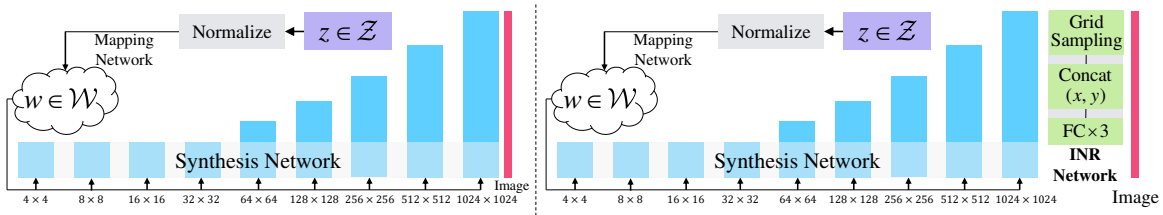
In this paper, we comprehensively diagnose perceptual losses for GAN inversion in an analysis-by-synthesis manner. We find some interesting properties for perceptual losses. First, we find that not down-sampling the input of perceptual losses is the key to recovering high-frequency details. For GAN inversion, it has been noticed that joint optimization of the latent code $\boldsymbol{w}$ and the noise maps $\boldsymbol{n}$ of StyleGAN will cause a lot of high-frequency artifacts [25]. Regularizing the noise maps alleviates the artifacts but compromises the reconstruction accuracy[1] (please see Fig. 1b and 1c). The VGG model is pre-trained on ImageNet [26] at 224^2 resolution. Thus previous methods usually down-sample the input image to 256^2 resolution before presenting the image to the VGG perceptual network [11], [27]. We find that the key to avoiding high-frequency artifacts while restoring image details lies in not down-sampling the input of the perceptual loss (see Fig. 1d).

On the other hand, no down-sampling causes a scale inconsistency problem (resolution mismatch) because the VGG model is pre-trained on images at low resolution but the input is at high resolution. No down-sampling leads to degraded PSNR compared to down-sampling. We find that the key to the scale inconsistency problem lies in which features to use to compute the perceptual loss. Specifically, the previous VGG perceptual loss uses the features of relu layers to calculate the loss, and the features are sparse. We find that dense features of convolutional layers are more robust to scale than the features of relu layers. Therefore, we propose to adopt convolutional features to calculate the perceptual loss.

Integrating these two designs, *i.e.*, no down-sampling and using convolutional features, derives our method, named **HRInversion**. We validate the effectiveness of HRInversion on a cross-domain image synthesis task. Specifically, the common inversion-based approach of cross-domain images synthesis contains two steps: (i) projecting real images into the latent space of GANs; and (ii) mixing the latent codes or models and then reconstructing cross-domain images. This approach is one-shot and the reconstructed images may be unsatisfactory. For example, the reconstructed image may be noisy and lose the semantics of the original image (see the examples in Fig. 2a, 2b, and 2c). To deal with this problem, we introduce an additional post-processing step, named local style optimization (LSO). LSO is based on the well-known semantic hierarchical nature of GANs: the layer-wise latent codes of GANs are specialized to different hierarchical semantics [29], [30]. For example, the early and middle layers determine pose and high-level semantics, and the later layers determine the color scheme. LSO fine-tunes only part of the latent codes to restore semantics and remove noise while keeping the style unchanged (see Fig. 2d). Note that our method belongs to optimization-based methods. Thus Fig. 2 summarizes three ways of mixing latent codes without considering encoder-based methods [31], [32]. The encoder-based methods adopt an encoder to extract the style code of the reference image and simultaneously optimize the generator and style-related branches during training. To quantitatively evaluate different algorithms, we introduce a quality measure named *ID retrieval* for face stylization (see Sec. III-D).

As shown in Fig. 1e, LSO synthesizes high-resolution face images with various styles, stably and controllably. We also diagnose the perceptual loss for GAN inversion layer by layer. The layer-wise analysis shows that low-level and high-level features play different roles for GAN inversion. Low-level features perform like pixel-wise losses, which induce blurry images. Although high-level features have lower spatial resolution than low-level features, they help recover high-frequency details such as hair texture. This again proves that the semantic space is superior to the pixel space in reconstructing high-fidelity images. Furthermore, we find that LSO benefits from a high-level perceptual loss discarding low-level features because LSO focuses more on semantic alignment rather than pixel alignment (see Fig. 14). It inspires us that we need to adjust the perceptual loss appropriately

[1] https://github.com/NVlabs/stylegan2-ada-pytorch



(a) StyleGAN2 (b) Style-INR-GAN

Fig. 3. The framework of Style-INR-GAN. (a) StyleGAN2 uses a mapping network to map the input noise $\boldsymbol{z}$ to the $\mathcal{W}$ space and then uses $w \in \mathcal{W}$ to modulate the feature maps of the synthesis network. (b) Compared to StyleGAN2, Style-INR-GAN is equipped with an INR network so that it can synthesize images of arbitrary resolution and aspect ratio. Therefore Style-INR-GAN is more suitable for inverting non-square images than StyleGAN2.

according to the task. For example, we can use a perceptual loss with only high-level features for semantic alignment or with only low-level features for pixel alignment.

We adopt StyleGAN2 [3] as the base model. To invert non-square images, we equip the generator with an implicit neural representation (INR) network so that the generator can generate images of arbitrary resolution and aspect ratio. HRInversion works well even for ultra-high resolution images (*e.g.*, panoramic images, up to 1677×6143 resolution). We also show applications of HRInversion in style transfer and 3D-aware GAN inversion. The results are impressive, validating its generalized ability and extending its range of applications. To facilitate people using HRInversion in their tasks, we provide a minimal implementation at our github open source site https://github.com/PeterouZh/HRInversion.

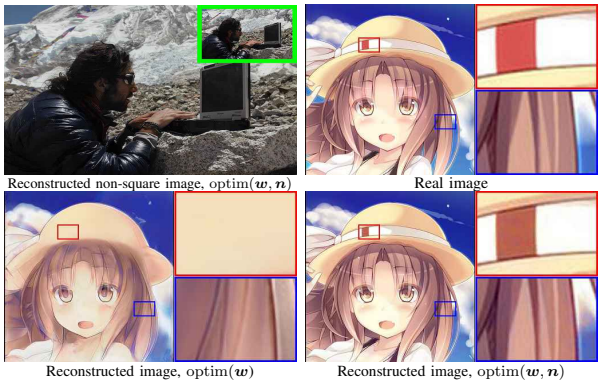


Fig. 4. Illustration of GAN inversion. Style-INR-GAN enables reconstruction of non-square images via GAN inversion. The latent code $\boldsymbol{w}$ is responsible for reconstructing low-frequency contours. The noise maps $\boldsymbol{n}$ enable the generation of high-frequency details. optim($\boldsymbol{w}$): only optimizing $\boldsymbol{w}$. optim($\boldsymbol{w}, \boldsymbol{n}$): optimizing $\boldsymbol{w}$ and $\boldsymbol{n}$ simultaneously. For the non-square image, the real image is highlighted with a green bounding box.

II. Related Work

Perceptual Metric. It is remarkably effective to use deep features of neural networks as a perceptual metric for image synthesis tasks [18], [27], [33]. For example, DeePSiM [33], computing distances between deep neural network features, enables a variational autoencoder to generate realistic high-resolution images. Zhang *et al*. [18] propose a metric named Learned Perceptual Image Patch Similarity (LPIPS) that agrees surprisingly well with human judgments. Samuli Laine [27] shows that a feature-based metric can produce more natural-looking interpolations than the norm-based metric in the image space. Deep features are also effective for style transfer [19]–[21], [34], [35]. Gatys *et al*. [19], [20] define a style reconstruction loss as the squared Frobenius norm of the difference between the Gram matrices computed by deep feature maps. Previous works usually adopt the perceptual loss as an off-the-shelf module. In this paper, we analyze the perceptual loss for the GAN inversion comprehensively. We demonstrate that no down-sampling for the input is critical to reconstruct image details for GAN inversion.

GAN Inversion. Based on GANs [36]–[39], there are two mainstream approaches for GAN inversion [40]: (i) learning an encoder to map a given image to the latent space; and (ii) optimizing a randomly initialized latent code using gradient descent. The encoder-based methods are fast but compromise image quality [16], [41]–[48]. The optimization-based methods are time-consuming but produce high-fidelity results [11], [12], [14], [15], [25], [29], [47], [49]–[58]. Inversion is an effective method for image manipulation and restoration [14]–[16]. For example, Abdal *et al*. [11], [25] propose to embed a given image into the latent space $\mathcal{W}^+$ of StyleGAN, enabling semantic image editing to existing photographs. PULSE [14] performs super resolution for face images by traversing the high-resolution natural image manifold modeled by a pre-trained generator of GANs. Chan *et al*. [16] propose an encoder-bank-decoder architecture to leverage rich priors encapsulated in a pre-trained GAN. In this paper, we leverage GAN inversion to do cross-domain image synthesis. We develop a post-processing step to improve the quality of the mixed image.

Image-to-Image Translation. Image-to-image translation aims to transfer input images from the source domain to the target domain [59]–[66]. However, creators are more concerned with creating images of new domains. In particular, given two images from source and target domains respectively, cross-domain image synthesis aims to merge this two images to generate an image of a novel domain. Style mixing [2] is an effective method to merge images from the same domain. However, because a generator can only generate images of a domain, style mixing cannot merge images from different domains. Layer swapping [28], [67] supports generating images in a new domain, but the synthesized images only have a single style and are susceptible to background noise. We aim to address these challenges: creating diverse styles of images while preserving semantic similarity of the source image.

III. Method

In this section, we first introduce a Style-INR-GAN model that can output images of arbitrary resolution and aspect

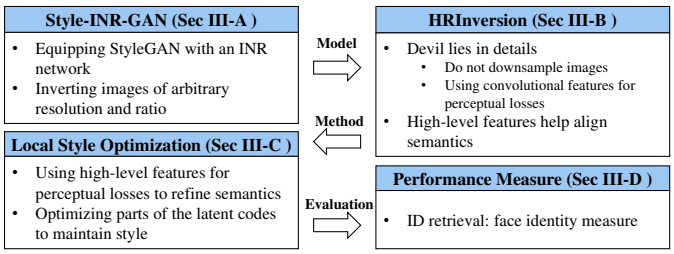


Fig. 5. The overall flowchart of our method. HRInversion combined with Style-INR-GAN has the ability to invert images of arbitrary resolution and aspect ratio. Based on our studies on perceptual losses, we propose a local style optimization algorithm capable of synthesizing clean and semantically preserved cross-domain images. We also introduce a quality measure named ID retrieval for face stylization algorithms.

ratio. We also present a mathematical form of GAN inversion (Sec. III-A). Based on the pretrained Style-INR-GAN, we study the influence of perceptual losses on GAN inversion in detail (Sec. III-B). We find that the devil lies in details for achieving satisfactory performance. We also observe that high-level features of VGG help align the semantics of images without degrading the background. Therefore, in Sec. III-C, we propose a post-processing algorithm that exploits high-level perceptual losses to refine the quality of synthesized cross-domain images. In Sec. III-D, we introduce a quality measure named ID retrieval for face stylization algorithms. To help readers quickly grasp the structure of the paper, we illustrate the overall flowchart of our method in Fig. 5.

A. Style-INR-GAN and GAN Inversion

Style-INR-GAN. As shown in Fig. 3, we remould StyleGAN2 [3] to enable it to generate images of arbitrary aspect ratio and size. We append an implicit neural representation (INR) network [68]–[70] after the synthesis network of StyleGAN2. Specifically, the INR network includes a grid sampling layer, which samples features at (x, y) coordinates from the output feature maps of StyleGAN2. Then, a concatenating layer concatenates features and coordinates together, followed by a 3-layer MLP to map the concatenated features to the RGB space. The INR network can output images of arbitrary aspect ratio and resolution, so it is suitable for GAN inversion to invert non-square images (see an example in Fig. 4). We name the model Style-INR-GAN, on which the experiments in this paper are based.

GAN Inversion. Given an image $\boldsymbol{x} \in \mathcal{X}$, GAN inversion aims to infer a latent code $\boldsymbol{z} \in \mathcal{Z}$, such that $G(\boldsymbol{z})$ and $\boldsymbol{x}$ are similar under a metric $d(\cdot)$ (G is a pre-trained generator). In this paper, we adopt $\mathcal{W}^+$ [11] space of StyleGAN2 because such space usually leads to high-quality inverted images. The inversion can be formulated as a minimization problem:

$$\min_{\boldsymbol{w},\boldsymbol{n}} d\left[G(\boldsymbol{w}+\boldsymbol{\epsilon},\boldsymbol{n}),\boldsymbol{x}\right] + \lambda_{\text{mse}} d_{\text{mse}}\left[G(\boldsymbol{w}+\boldsymbol{\epsilon},\boldsymbol{n}),\boldsymbol{x}\right], \quad (1)$$

where $\boldsymbol{w} \in \mathcal{W}^+$ can be initialized with the average latent code $\bar{\boldsymbol{w}}$. $\boldsymbol{n}$ denotes the noise variables of StyleGAN2, encoding high-frequency details [25]. $\boldsymbol{\epsilon}$ is a random Gaussian noise to facilitate optimization. Its dimension is equal to $\boldsymbol{w}$ and variance gradually decreases with iterations [4], [71]. $d(\cdot)$ denotes a metric function measuring the difference between the synthesized image and the real image. $d_{\text{mse}}(\cdot)$ means pixel-wise MSE loss.

Eq. (1) can be solved by gradient descent, which usually takes a few minutes for an image. We provide an example of GAN inversion to illustrate the role of $\boldsymbol{w}$ and $\boldsymbol{n}$. As shown in Fig. 4, it is apparent that $\boldsymbol{w}$ is responsible for representing low-frequency contours and $\boldsymbol{n}$ is for high-frequency details. $\boldsymbol{w}$ is more editable than $\boldsymbol{n}$ because $\boldsymbol{w}$ encodes more semantics than details.

B. HRInversion: The Devil Lies in Details

Relu-based Perceptual Loss. GAN inversion usually adopts perceptual losses as the metric function. In contrast to MSE, perceptual losses utilize a pre-trained VGG model to extract features and compute the loss in the feature space. Previous commonly used perceptual losses [11], [21] extract features from relu layers, which are sparse. Besides, most GAN inversion methods usually down-sample the input to 256^2 resolution before passing the high-resolution images through the VGG network [11], [25], [27]. Thus the VGG relu-based perceptual loss is given by:

$$d^{(\text{relu})}(\hat{\boldsymbol{x}},\boldsymbol{x}) = \sum_l ||\lambda_l \left[\phi_l^{\text{relu}}\left(\varphi(\hat{\boldsymbol{x}})\right) - \phi_l^{\text{relu}}\left(\varphi(\boldsymbol{x})\right)\right]||_2^2, \quad (2)$$

where $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ are the reconstructed and target image, respectively. $\phi_l^{\text{relu}}(\cdot)$ denotes the feature maps of the lth relu layer of VGG. λ_l is the coefficient for the lth layer. $\varphi(\cdot)$ represents the down-sampling operation. The reason for down-sampling is that VGG [72] is pre-trained on ImageNet [26] at 224^2 resolution. However, the images to be inverted are usually of higher resolution (*e.g.*, 1024^2 in this paper). Such scale inconsistency will deteriorate PSNR (see Tab. II).

Conv-based Perceptual Loss. It has been noticed that jointly optimizing the latent code $\boldsymbol{w}$ and the noise maps $\boldsymbol{n}$ results in a lot of high-frequency artifacts [25] (see Fig. 1b). We find that the key to eliminating artifacts and restoring image details lies in not down-sampling the input of the perceptual loss. As shown in Fig. 1d, no down-sampling allows GAN inversion to restore the details of the original image while avoiding artifacts. Moreover, we also test other perceptual losses such as ResNet and transformer-based perceptual losses. No down-sampling also yields better results (please refer to Sec. IV-B for details).

On one hand, for high-resolution GAN inversion, if we down-sample the input, it will cause high-frequency artifacts on the reconstructed image. On the other hand, if we do not down-sample the input, the perceptual loss suffers from a scale inconsistency problem (thus yielding suboptimal PSNR) because the perceptual network is pre-trained on low-resolution images. We find that features from VGG convolutional layers are more robust to scale than features from relu layers. To avoid high-frequency artifacts while attenuating the influence of scale, we propose a perceptual loss named HRInversion which does not down-sample the input and computes the perceptual loss using features of convolutional layers.

For HRInversion, the pre-trained VGG model comes from the timm library [73]. We carefully selected five convolutional

layers whose names are features_2, features_7, features_14, features_21, and features_28, respectively, covering both low-level and high-level layers. Thus our VGG conv-based perceptual loss is defined as

$$d^{(\text{conv})}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \sum_{l \in \{2,7,14,21,28\}} ||\lambda_l \left[\phi_l^{\text{conv}}(\hat{\boldsymbol{x}}) - \phi_l^{\text{conv}}(\boldsymbol{x})\right]||_2^2, \tag{3}$$

where $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ are the reconstructed and target image, respectively. $\phi_l^{\text{conv}}(\cdot)$ denotes the feature maps of the lth convolutional layer of the pre-trained VGG. λ_l are set to be 0.0002, 0.0001, 0.0001, 0.0002, and 0.0005, empirically.

High-level Perceptual Loss. We study the nature of perceptual losses in detail through extensive experiments (see Sec. IV-G). Layer-wise diagnosis of perceptual losses enables us to discover that the high-level features of perceptual models concentrate on semantic details rather than pixels. This prompts us to propose a high-level perceptual loss, $d_h(\cdot)$, which is dedicated to semantic alignment rather than pixel alignment. The high-level perceptual loss is defined as:

$$d_h(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \sum_{l \in \{21,28\}} ||\lambda_l \left[\phi_l^{\text{conv}}(\hat{\boldsymbol{x}}) - \phi_l^{\text{conv}}(\boldsymbol{x})\right]||_2^2, \tag{4}$$

where $d_h(\cdot)$ only uses two top convolutional layers of VGG so that it can align images semantically without degrading the background. In what follows, we propose a post-processing algorithm that exploits the high-level perceptual loss to improve the quality of cross-domain images.

C. Cross-domain Face Synthesis

Given two face images $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ from source and target domains respectively, we aim to synthesize an image of a novel domain. The common inversion-based method consists of two steps: (i) projecting the images into the latent space; and (ii) mixing latent codes or models. This method is susceptible to noise and at the risk of losing the semantics of the source image. In what follows, we propose a post-processing step to deal with these issues.

a) ***Projecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ into the $\mathcal{W}^+$ space:*** According to Eq. (1), we get

$$\begin{aligned}
\boldsymbol{w}_s &= \min_{\boldsymbol{w}} d^{(\text{conv})}\left[G_s(\boldsymbol{w}+\boldsymbol{\epsilon}), \boldsymbol{x}_s\right] + \lambda_{\text{mse}} d_{\text{mse}}\left[G_s(\boldsymbol{w}+\boldsymbol{\epsilon}), \boldsymbol{x}_s\right], \\
\boldsymbol{w}_t &= \min_{\boldsymbol{w}} d^{(\text{conv})}\left[G_t(\boldsymbol{w}+\boldsymbol{\epsilon}), \boldsymbol{x}_t\right] + \lambda_{\text{mse}} d_{\text{mse}}\left[G_t(\boldsymbol{w}+\boldsymbol{\epsilon}), \boldsymbol{x}_t\right],
\end{aligned} \tag{5}$$

where $\boldsymbol{w} \in \mathcal{W}^+$ [11] is the latent code to be solved, and $\boldsymbol{\epsilon}$ is a random Gaussian noise to facilitate optimization. $d^{(\text{conv})}(\cdot)$ is our conv-based perceptual loss defined in Eq. (3). G_s and G_t are generators trained on source and target domain where the G_t is fine-tuned from G_s. Note that we do not use noise variables $\boldsymbol{n}$ because it is less editable than the latent code $\boldsymbol{w}$.

b) ***Blending parameters and latent codes:*** We adopt linear interpolation to mix generators and latent codes, respectively. The mixing methods are given by

$$G_m = \text{lerp}(G_s, G_t, \lambda) = G_s + \lambda(G_t - G_s), \tag{6}$$

$$\boldsymbol{w}_m = \text{lerp}(\boldsymbol{w}_s, \boldsymbol{w}_t, \lambda) = \boldsymbol{w}_s + \lambda(\boldsymbol{w}_t - \boldsymbol{w}_s), \tag{7}$$

where $0 \leq \lambda \leq 1$ is the coefficient of linear interpolation. $\text{lerp}(G_s, G_t)$ means to do linear interpolation for all parameters of the generators. By default, λ is set to 0.2 for b4-b64 and 0.7 for b128-b1024. The same setting applies to the λ of $\boldsymbol{w}$. Note that layer swapping is a special case of Eq. (6), where it swaps part of the generator parameters according to the resolution level [28]. Given a source image $\boldsymbol{x}_s$, layer swapping only synthesizes images of a single style because the latent code $\boldsymbol{w}_s$ is fixed (please see Fig. 2a). If we apply layer swapping to latent codes as well, we can get various styles of images, as shown in Fig. 2b. In this paper, we adopt linear interpolation as the mixing method because it is flexible for adjusting the degree of mixing.

Due to the outstanding nature of the StyleGAN, the reconstructed image $G_m(\boldsymbol{w}_m)$ is probably already on the natural image manifold. However, $G_m(\boldsymbol{w}_m)$ may lose the pose and semantics of the source image, $\boldsymbol{x}_s$, as shown in Fig. 2c. To make matters worse, $G_m(\boldsymbol{w}_m)$ may hallucinate some visual noise that do not exist in $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$, leading to visually disappointing results. We hope to obtain clean and semantics-preserved images of a new domain. Therefore, a post-processing step is required.

c) ***Post-processing by local style optimization:*** $G_m(\boldsymbol{w}_m)$ already contains some semantic attributes of $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$. We hope to amend $\boldsymbol{w}_m$ so that $G_m(\boldsymbol{w}_m)$ maintains current style, restores the lost pose and semantics of $\boldsymbol{x}_s$, and removes noise. This is achieved by a post-processing step. In particular, let $\boldsymbol{w}_m$ be the base latent code. We introduce a new optimization variable $\boldsymbol{w}_{\text{opt}}$, which is initialized to small random numbers around zero. The objective function is defined as

$$\min_{\boldsymbol{w}_{\text{opt}}, \boldsymbol{n}} d_h\left[G_m(\boldsymbol{w}_m + \boldsymbol{w}_{\text{opt}} + \boldsymbol{\epsilon}, \boldsymbol{n}), \boldsymbol{x}_s\right] + \lambda_{\text{reg}} ||\boldsymbol{w}_{\text{opt}}||_2^2, \tag{8}$$

where $\boldsymbol{w}_m \in \mathcal{W}^+$ is the mixed latent code and keeps fixed. $\boldsymbol{w}_{\text{opt}}$ and $\boldsymbol{n}$ are variables to be optimized. $\boldsymbol{n}$ are the noise variables of StyleGAN2. λ_{reg} is the coefficient of L2 regularization. $d_h(\cdot)$ is the high-level perceptual loss defined in Eq. (4), which only uses the *high-level convolutional features* of the VGG network. As a result, $d_h(\cdot)$ focuses on semantic alignment rather than pixel-wise alignment, which is critical for restoring semantics while avoiding deteriorating background (please see Sec. IV-G for details). We optimize both $\boldsymbol{w}_{\text{opt}}$ and $\boldsymbol{n}$ using the same Adam optimizer. $\boldsymbol{w}_{\text{opt}}$ is to restore pose and semantics, and $\boldsymbol{n}$ to finer details. We empirically find that the contribution of $\boldsymbol{n}$ is negligible and the main contribution comes from $\boldsymbol{w}_{\text{opt}}$. In the following, we omit notation $\boldsymbol{n}$ for simplicity.

We adopt two constraints to ensure that $\boldsymbol{w}_m + \boldsymbol{w}_{\text{opt}}$ is always around $\boldsymbol{w}_m$. First, we only choose partial latent codes instead of all to optimize, named selective optimization. In particular, $\boldsymbol{w}_m$ contains a total of 18 latent codes for StyleGAN2 at 1024×1024 resolution. It is acknowledged that the early and middle layers control the pose and semantic attributes of the generated images, and the last layers control the color scheme [29], [30], [75]. Therefore, we can only optimize latent codes of early layers (*e.g.*, $\{\boldsymbol{w}_{\text{opt}}^i | i = 0, 1, 2, 3, 4\}$) to adjust the pose and semantics of $G_m(\boldsymbol{w}_m + \boldsymbol{w}_{\text{opt}})$ while keeping the current style unchanged.

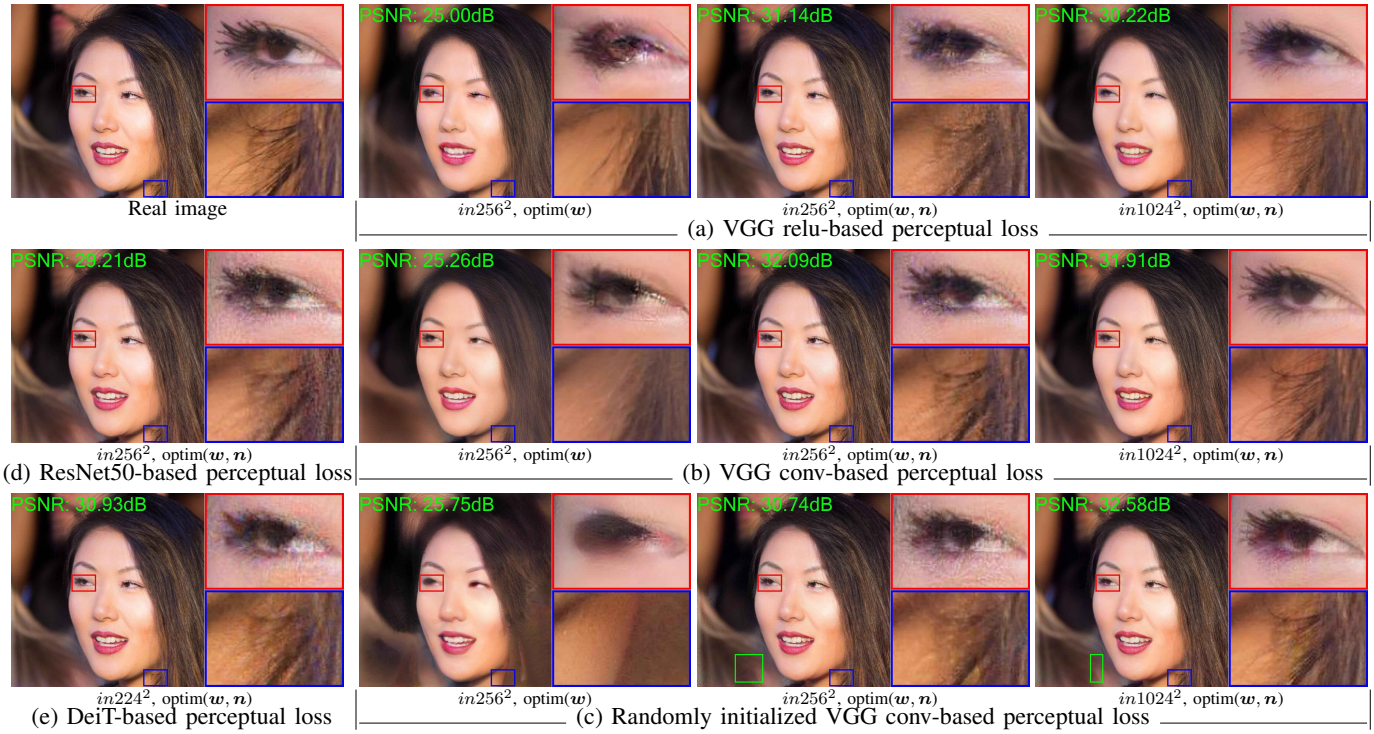


Fig. 6. Jointly optimizing $\boldsymbol{w}$ and $\boldsymbol{n}$ (*i.e.*, optim($\boldsymbol{w}, \boldsymbol{n}$)) recovers more detail, but suffers from high-frequency artifacts when down-sampling the input of perceptual losses. No down-sampling eliminates high-frequency artifacts but deteriorates PSNR (see (a) and (b)) for the pretrained VGG perceptual network. Note that the VGG conv-based perceptual loss is less affected by the scale shift issue than the VGG relu-based perceptual loss, namely $32.09-31.91(0.18dB\downarrow)$ *vs.* $31.14-30.22(0.92dB\downarrow)$. Interestingly, the randomly initialized VGG does not suffer from scale shift problem but it induces non-smooth artifacts, as shown in the green bounding boxes in (c). Please zoom in to see details. (e) DeiT [74] is a pre-trained transformer model whose input is at 224×224 resolution, suffering from lots of artifacts. $in256^2$: input at 256^2 resolution. Please refer to Sec. IV-B for details.

Second, we apply L2 regularization to $\boldsymbol{w}_{\text{opt}}$, which ensures that $\boldsymbol{w}_m+\boldsymbol{w}_{\text{opt}}$ does not become too specialized to cause overfitting. Selective optimization and L2 regularization ensure that $\boldsymbol{w}_m+\boldsymbol{w}_{\text{opt}}$ is always around $\boldsymbol{w}_m$, so as to retain the current style of $G_m(\boldsymbol{w}_m)$. Meanwhile, optimizing partial latent codes helps restore the pose and semantics of the source image. Therefore, we call this step local style optimization (LSO). LSO is a post-processing step that does not require too many iterations. In practice, it is efficient and achieves stable results with only 50 iterations.

D. Performance Measure

Before presenting the experiments, we introduce a quality measure for face stylization approaches. Most related works adopt FID between generated images and style images to measure the performance of the algorithm. However, we consider it insufficient as it cannot assess the similarity of face identities between content and stylized images. If the stylized image loses the face identity of the content image, we consider the result meaningless. To evaluate whether the stylized image is consistent with the content image in terms of face identity, inspired by FaceShifter [76], we propose a metric named ID retrieval. The details of ID retrieval are as follows.

ID retrieval. We select the first 100 images of CelebA-HQ [77] as content images, which are not seen by the model during training. We randomly synthesize 50 stylized images for each content image, so there are 5000 stylized images in total. We extract face identity vectors for content and stylized images with a pre-trained face recognition network [78]. For each stylized image, we search for its nearest face in the content images and check if the nearest face matches the original content face. The distance adopts the Euclidean distance between the face identity vectors. ID retrieval is the accuracy rate equal to the proportion of successfully matched images to all stylized images. To facilitate others using this criterion to evaluate their algorithms, we have released the ID retrieval implementation at https://github.com/PeterouZh/ID_retrieval.

IV. Experiments

We perform extensive experiments to verify the effectiveness of our method. First, we point out that no down-sampling is the key to avoiding high-frequency artifacts for GAN inversion, and propose to employ convolutional features to calculate the perceptual loss to reduce the impact of scale inconsistency (see Sec. IV-B). Second, we quantitatively study several perceptual losses and find that relu features are indeed more sensitive to scale than convolutional features (see Sec. IV-C). In Sec. IV-D, we show that LSO is an effective post-processing step to remove noise and preserve style qualitatively. In Sec. IV-E, we adopt ID retrieval to quantitatively evaluate our method. Finally, we show applications of HRInversion for high-resolution panoramic images and 3D-aware GAN inversion (see Sec. IV-F). For reproduction purpose, we will release code and pre-trained models at our github open source site https://github.com/PeterouZh/HRInversion.

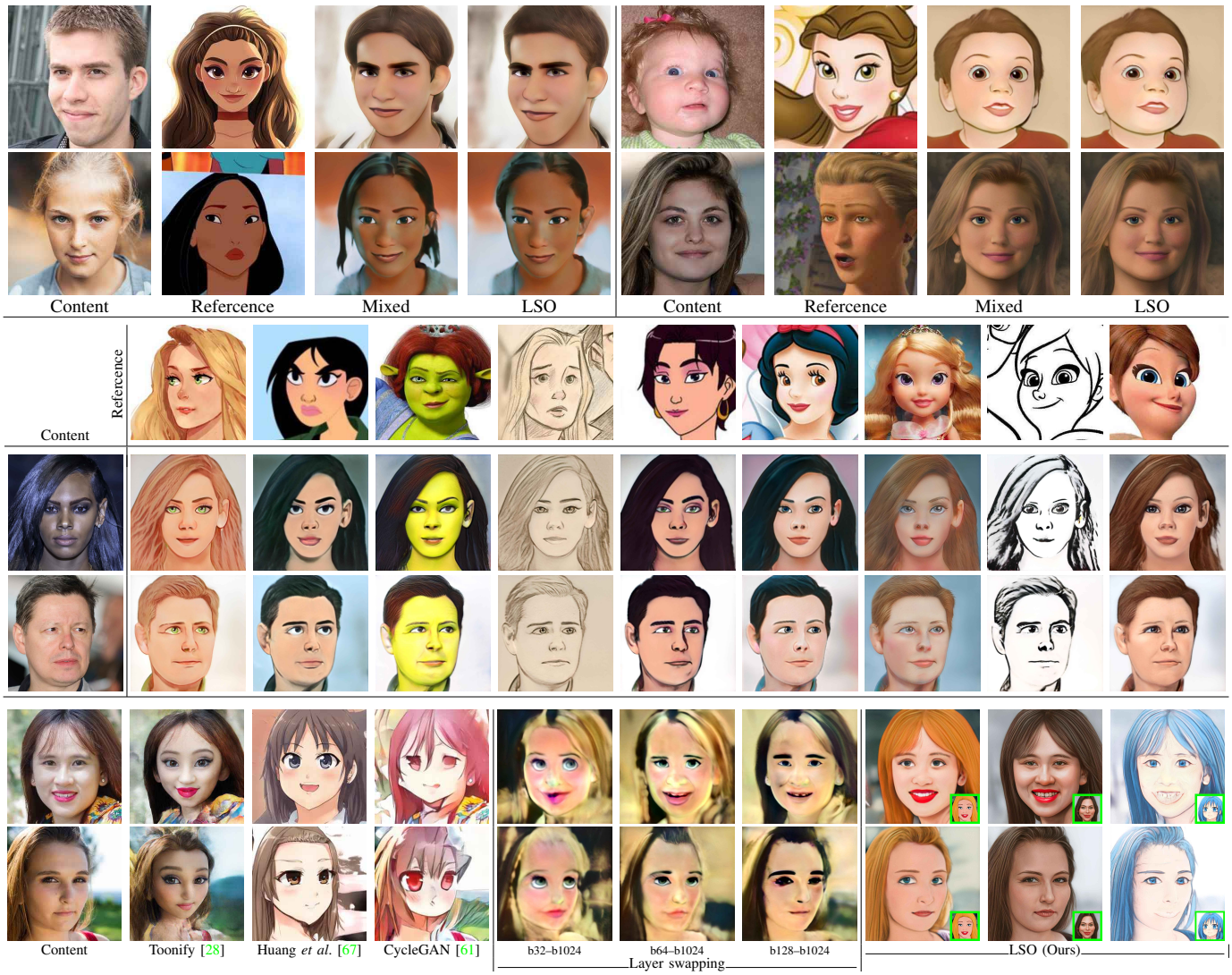


Fig. 7. Top: LSO helps restore semantics of the content image, align the pose, and remove noise existing in the mixed images. More importantly, LSO does not change the current style of the mixed image. Middle: Our method is capable of synthesizing images of various styles for a content image. Bottom: Other methods lose the semantics of the content and suffer from disturbing noise. In contrast, our method supports different styles while preserving more face identities. b32-b1024: swap the blocks of the source generator from block b32 to b1024 with the corresponding blocks of the target generator.

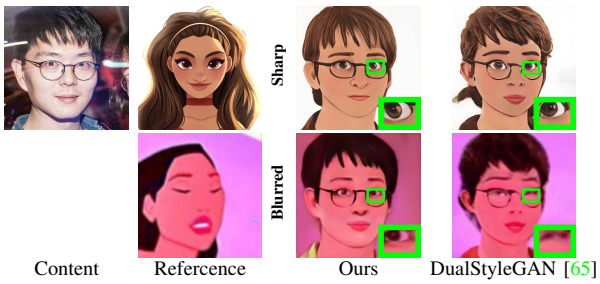


Fig. 8. As a reference-based method, the quality of the stylized images synthesized by our method is influenced by the quality of the reference image. As demonstrated in the bottom images, blurred reference images may deteriorate the quality of the stylized images. This phenomenon also exists for DualStyleGAN [65].

A. Experimental Details

We first train Style-INR-GAN on FFHQ [2] for $25,000$k images with the default setting of StyleGAN2-ADA [4]. FFHQ contains $70,000$ face images at 1024×1024 resolution. We augment the training images with horizontal flipping. The model pre-trained on FFHQ is regarded as the source model. We employ transfer learning to obtain the target model, where the source model is fine-tuned on the target domain dataset containing 317 Disney-style images [28]. We fine-tune the model for only 1000k images with ADA augmentation as fine-tuning converges faster than training from scratch. The pre-trained perceptual models come from the timm library [73].

Tab. I presents more details of the method. We deliver two messages. First, the training time of Style-INR-GAN is slightly longer than that of StyleGAN2 (7d 20h *vs.* 6d 3h, using 8*V100 GPUs) because Style-INR-GAN has an additional INR sub-network compared to StyleGAN2. Style-INR-GAN is slightly worse than StyleGAN2 in terms of FID (3.26 *vs.* 2.94). The difference may be because we did not employ path length regularization [3] during training. Nonetheless, Style-INR-GAN is more flexible than StyleGAN2 because it supports inverting non-square images. Second, our face stylization method belongs to optimization-based methods. Given a content image, we need to project it into the latent space of GANs. Therefore, its time efficiency is worse than encoder-based methods. However, our method outperforms encoder-

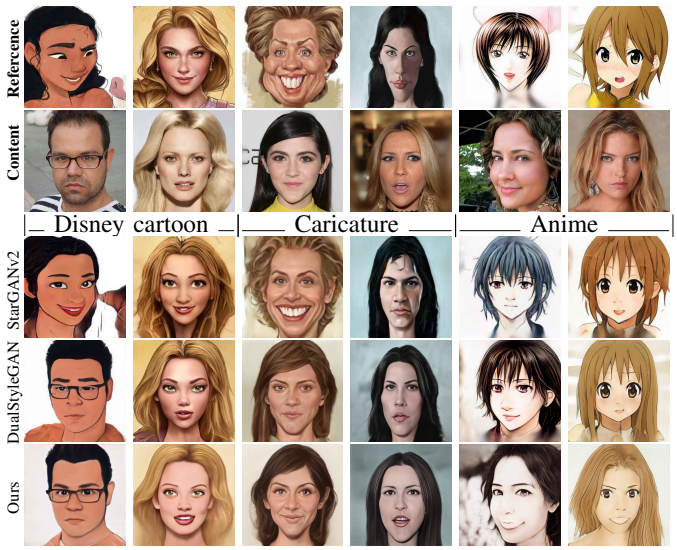


Fig. 9. Visual comparison of reference-based face stylization. StarGANv2 overfits the reference images and ignores the content images. DualStyleGAN is good at transferring style but not at preserving face identities, especially for the Anime style. The strength of our method lies in preserving face identities.

TABLE I
DETAILS OF TRAINING AND INVERSION.

Training (FFHQ)	Resolution	GPUs	Batch size	GPU mem	Time (25,000 kimg)	FID
StyleGAN2 [3]	1024^2	8	32	8.3 GB	6d 03h	2.94
Style-INR-GAN				8.9 GB	7d 20h	3.26
Style-INR-GAN (fine-tuning on disney)	1024^2	8	32	8.9 GB	8h 43m (1,000 kimg)	17.8

	Resolution	GPUs	Batch size	GPU mem	Iterations	Time
Inversion	1024^2	1	1	5.89 GB	1,000	2m 18s
LSO				5.38 GB	50	8s
Generation				0.78 GB	-	32ms

TABLE II
TESTING PERCEPTUAL LOSSES ON CELEBA-HQ AT 1024×1024 RESOLUTION WITH 1000 ITERATIONS FOR GAN INVERSION. NO DOWN-SAMPLING CONSISTENTLY IMPROVES THE PERCEPTUAL METRIC, LPIPS. PLEASE REFER TO SEC. IV-C FOR DETAILS.

	Method	Input resolution	PSNR ↑ (dB)	LPIPS ↓
optim($\boldsymbol{w}$)	Image2StyleGAN [11]	256×256	24.36	0.4226
optim($\boldsymbol{w}, \boldsymbol{n}$)	Image2StyleGAN++ [25]	256×256	**30.44**	**0.3590**
Relu-based	LPIPS [18]	256×256	**31.69**	0.2587
	LPIPS	1024×1024	31.24	**0.0591**
	VGG_relu	256×256	**31.04**	0.3533
	VGG_relu	1024×1024	30.62	**0.0523**
Conv-based	VGG_conv	256×256	31.62	0.3394
	VGG_conv	1024×1024	**33.16**	**0.0703**
	ResNet50	256×256	32.14	0.3285
	ResNet50	1024×1024	**32.87**	**0.1346**

based methods in preserving face identities of content images. In Sec. IV-E, we adopt *ID retrieval* to quantitatively compare our method with encoder-based methods (StarGANv2 [63] and DualStyleGAN [65]). We find that current encoder-based methods are still unsatisfactory in preserving face identities.

B. Perceptual Losses: Diagnostic Studies

We investigate the properties of several perceptual losses:

TABLE III
QUANTITATIVE COMPARISON OF REFERENCE-BASED FACE STYLIZATION. HIGHER ID RETRIEVAL INDICATES THAT THE STYLIZED IMAGES RETAIN MORE FACE IDENTITIES. LOWER FID DOES NOT NECESSARILY IMPLY A BETTER APPROACH BECAUSE THE STYLIZED IMAGE MAY HAVE LOST THE FACE IDENTITY OF THE CONTENT IMAGE. WE PROVIDE QUALITATIVE RESULTS AND EXPLANATIONS IN FIG. 9.

Datasets	Methods	ID Retrieval↑	FID↓ (1024^2)	FID↓(256^2)
Disney cartoon [28]	StarGANv2 [63]	1.06%	-	39.35
	DualStyleGAN [65]	11.98%	**23.72**	**25.91**
	Ours	**60.30%**	56.80	65.29
Caricature [79], [80]	StarGANv2	0.98%	-	37.03
	DualStyleGAN	21.64%	**22.54**	**25.56**
	Ours	**80.76%**	48.71	49.65
Anime [81]	StarGANv2	1.10%	-	40.13
	DualStyleGAN	1.34%	**12.63**	**14.29**
	Ours	**7.52%**	53.81	58.62

*a) **Perceptual loss without down-sampling**:* Abdal *et al.* [25] found that the VGG relu-based perceptual loss is incompatible with noise maps $\boldsymbol{n}$. For example, Fig. 6a (middle) suffers from lots of high-frequency artifacts. We find that the reason for high-frequency artifacts is that previous methods generally down-sample the input of 1024×1024 resolution to 256×256 resolution. No down-sampling eliminates high-frequency artifacts, achieves better perception quality, but deteriorates PSNR, as shown in Fig. 6a (right). We refer to this phenomenon as a scale inconsistency problem [11], [27] in that the VGG model is trained for images at lower resolution. We also have tested other types of perceptual networks such as ResNet50 [82] and DeiT [74] (*cf*. Fig. 6d and 6e). DeiT [74] is a pre-trained transformer model whose input is at 224×224 resolution. We found that down-sampling the input consistently leads to high-frequency artifacts.

*b) **VGG conv-based perceptual loss**:* We notice that the previous VGG perceptual loss usually uses the feature maps of relu layers. The output feature maps of relu are sparse, where most of the elements are zero. This limits the gradients of the backpropagation, which may limit the perceptual ability [83]. Fig. 6b shows the results of the VGG conv-based perceptual loss [2]. For VGG conv-based perceptual loss, down-sampling also causes high-frequency artifacts. The reconstructed image without down-sampling is of high perception quality and its PSNR is better than that of the VGG relu-based perceptual loss (Fig. 6b *vs.* Fig. 6a). Nevertheless, no down-sampling deteriorates the PSNR for the VGG conv-based perceptual loss. However, the VGG conv-based perceptual loss is less affected by the scale inconsistency issue than the VGG relu-based perceptual loss, namely $32.09 - 31.91(0.18dB \downarrow)$ *vs.* $31.14 - 30.22(0.92dB \downarrow)$.

*c) **Randomly initialized VGG perceptual loss**:* Liu *et al.* [84] claimed that using a randomly initialized network as the perceptual model could achieve a similar effect to the pre-trained network. The randomly initialized network is scale-independent because the weights of the network have not been trained on images at a specific resolution. Fig. 6c shows the reconstructed images obtained using a randomly initialized VGG

[2]Be careful that the convolutional layer and the relu layer share the same output memory by default in the PyTorch VGG implementation (*i.e.*, the *inplace* argument of relu is true).

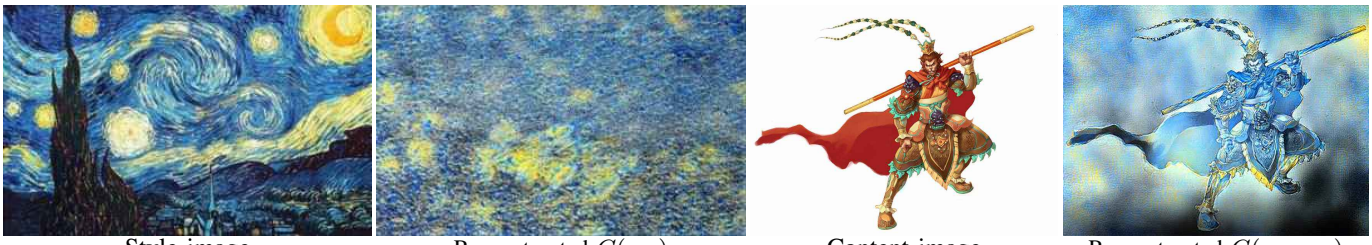
(a) Real image

(b) HRInversion result (at 441 × 2048 resolution)

(c) Real image

(d) HRInversion result (at 1677 × 6143 resolution)

Fig. 10. HRInversion results for ultra-high resolution images with arbitrary aspect ratio. We adopt the generator of Style-INR-GAN because it can output images of arbitrary resolution and aspect ratio, and thus is friendly to inverting non-square images. The perceptual loss adopts the features of the pre-trained VGG convolutional layers and the input is not down-sampled.



Style image | Reconstructed $G(\boldsymbol{w}_s)$ | Content image | Reconstructed $G(\boldsymbol{w}_s, \boldsymbol{n}_c)$

Fig. 11. An example of style transfer using HRInversion. Given a style image, we use GAN inversion to get the latent code $\boldsymbol{w}_s$. The noise maps $\boldsymbol{n}$ use the default noise maps of the generator and remain fixed. The reconstructed image $G(\boldsymbol{w}_s)$ cannot recover the details of the original image because the style image is not on the manifold of the training data. For the content image, we initialize the latent code with $\boldsymbol{w}_s$, keep the latent code unchanged, and only optimize noise maps. Because noise maps have the ability to reconstruct image details, the reconstructed image $G(\boldsymbol{w}_s, \boldsymbol{n}_c)$ achieves the effect of style transfer.

conv-based perceptual loss. Down-sampling still causes high-frequency artifacts. To our surprise, the reconstructed image without down-sampling (Fig. 6c (right)) are even better than those of the pre-trained models in terms of PSNR. However, although the randomly initialized network has a higher PSNR, it causes a lot of non-smooth artifacts. Please zoom in to see the artifacts in the green bounding boxes of Fig. 6c. In summary, the properties of randomly initialized models and pre-trained models are different. Randomly initialized models are not specific to images at a specific resolution. However, randomly initialized models will induce non-smooth artifacts, deteriorating human perception quality.

C. Perceptual Losses: Quantitative Results

To quantitatively evaluate different perceptual losses, we adopt CelebA-HQ [77] at 1024^2 resolution as the testbed and use PSNR and the perceptual metric, LPIPS [18], to measure the similarity between the original images and the reconstructions. Table II shows the results. First, jointly optimizing latent code $\boldsymbol{w}$ and noise maps $\boldsymbol{n}$ significantly improves PSNR (Image2StyleGAN *vs.* Image2StyleGAN++). Second, relu-based perceptual losses are more sensitive to scale than conv-based perceptual losses. For example, the LPIPS and VGG_relu achieve worse PSNR at 1024^2 resolution than at 256^2 resolution. In contrast, both VGG_conv and ResNet50 achieve better PSNR at 1024^2 resolution than at 256^2 resolution. Third, no down-sampling consistently improves the perceptual metric, LPIPS. This is in line with human perception, as down-sampling causes artifacts despite potentially high PSNR. Last but not least, we find that LPIPS is a biased metric. VGG-based perceptual losses help to improve the LPIPS metric. For example, the VGG-based perceptual losses (LPIPS, VGG_relu, and VGG_conv) achieve better LPIPS than the ResNet-based perceptual loss at 1024^2 resolution. It is reasonable because LPIPS is computed based on the VGG model.

D. Cross-Domain Image Synthesis: Qualitative Results

We validate the effectiveness of local style optimization (LSO). LSO is a post-processing step that provides us with an opportunity to repair the initially unsatisfactory mixed images. As shown at the top of Fig. 7, LSO effectively restores the semantics of the input image for the mixed image, aligns the pose, and removes noise existing in the mixed images. These effects are completed without changing the style of the mixed images. Since we mix generator parameters and latent codes simultaneously, we can synthesize images of various styles for an input image by constructing a style gallery. Fig. 7 (middle) shows a number of different styles of images synthesized by our method.

Fig. 7 (bottom) presents some results for other methods. The problem of other methods is that they are one-shot and

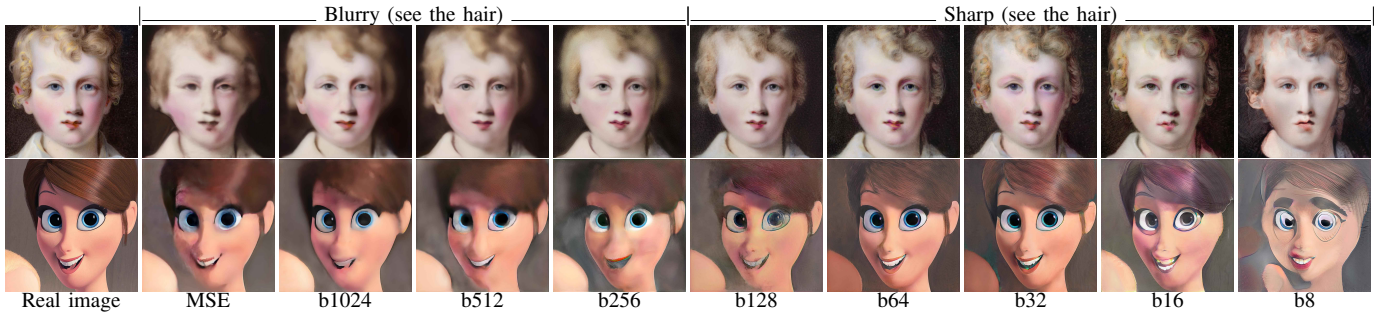


Fig. 12. Layer-by-layer analysis for the perceptual loss. MSE is a pixel-wise loss and the reconstructed image tends to be blurry. The low-level features of the perceptual loss (from block b1024 to b256) play a role similar to MSE, and neither can reconstruct the details of the hair. The high-level features (from block b128 to b8) help restore high-frequency details such as the hair texture. b1024 denotes that the features are at 1024^2 resolution.

thus uncontrollable. It is possible to lose the semantics of the input image and suffer from a lot of noise. We also show the results of layer swapping, which are not very satisfactory. We think there are two problems with layer swapping: (i) it is not easy to determine at which layer to start the swapping; and (ii) it only exchanges the weights of the last layers and ignores the weights of the early layers, resulting in incomplete style fusion compared to interpolating all the weights. Although interpolating all weights may change the semantics of the mixed image, LSO alleviates this problem by finetuning the early latent codes.

Our method belongs to reference-based style transfer, and the quality of the stylized images are affected by the quality of the reference images. As shown in Fig. 8, when the reference image is sharp, the stylized image is also sharp. If the reference image is of poor quality, the stylized image becomes blurred. This phenomenon also exists for DualStyleGAN [65], the SOTA method of face toonification.

E. Cross-Domain Image Synthesis: Quantitative Results

We compare with state-of-the-art methods StarGANv2 [63] and DualStyleGAN [65] on three datasets Disney cartoon [28], Caricature [79], [80], and Anime [81]. The data processing pipeline follows DualStyleGAN. We use ID retrieval to assess how well the algorithm retains face identities. We also provide FID to measure the distance between the distribution of stylized images and the reference images. Note that the default resolution of StarGANv2 is 256×256. We trained StarGANv2 with the official default settings, so we only provide results at 256×256 resolution for StarGANv2. Nevertheless, the difference in resolution does not affect the ID retrieval criterion because the images will be resized to 112×112 before being passed to the face recognition model.

As shown in Tab. III, the ID retrieval values of StarGANv2 are close to the random value of 1% on all three datasets (the total number of content images is 100, so the random value for ID retrieval is 1%). This indicates that the synthesized images of StarGANv2 completely lose the face identities of the content images. DualStyleGAN achieves the best FID on all three datasets, indicating that the distribution of images synthesized by DualStyleGAN is close to that of the reference images. However, the ID retrieval of DualStyleGAN is inferior (*e.g.*, 11.98% on Disney cartoon and 21.64% on Caricature). To make matters worse, the ID retrieval of DualStyleGAN on Anime is only 1.34%, which is close to random. ID retrieval quantitatively reveals the flaw that DualStyleGAN may lose the face identities of content images.

In contrast, our method consistently achieves the best ID retrieval on all three datasets. Nonetheless, the FID metric of our method remains to be improved. Our results indicate that there is a trade-off between ID retrieval and FID. We still look forward to better methods for this field to improve both metrics simultaneously. Fig. 9 shows the qualitative results. It is evident that StarGANv2 overfits the reference images and ignores the content images. The results of DualStyleGAN are similar in style to the reference images, but lose the face identities of the content images, especially for the Anime style. Our results take into account both preserving the face identities of the content images and transferring the style of the reference images.

Overall, our results reveal the properties of current methods. For example, StarGANv2 nearly completely ignores the face identity of the content image. DualStyleGAN is good at transferring style but not at preserving face identities. The strength of our method lies in preserving face identities. We have released the evaluation code of ID retrieval. We believe it will facilitate future research.

In addition, we also use the perceptual metric, LPIPS, to measure the similarity between the stylized and the content images. We select 70 real images from CelebA-HQ and synthesize 187 stylized images for each real image. Therefore, there are $13,090$ stylized images in total. As shown in Fig. 13, LSO improves LPIPS for most images. Some exceptions appear at the points of large LPIPS (highlighted by the blue ellipse), where the semantics between the mixed and input images are quite different. We conjecture the reason is that LSO is a locally optimizing approach, so it may fail when the semantic gap is large. Overall, LSO is an effective post-processing technique for most cross-domain images.

F. Applications of HRInversion

To reconstruct high-definition images while avoiding the scale inconsistency problem, we propose HRInversion which uses convolutional features for the perceptual loss without down-sampling the input. We verify the effectiveness of HRInversion on ultra-high resolution panoramas and 3D-aware GAN inversion respectively.

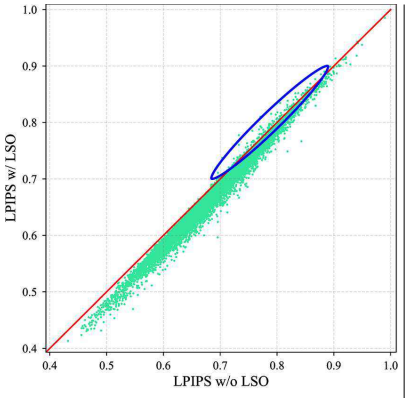


Fig. 13. LSO improves the perceptual metric, LPIPS. Each dot corresponds to a cross-domain image. After LSO, most of the images have been improved in terms of LPIPS (most points are below the red diagonal).

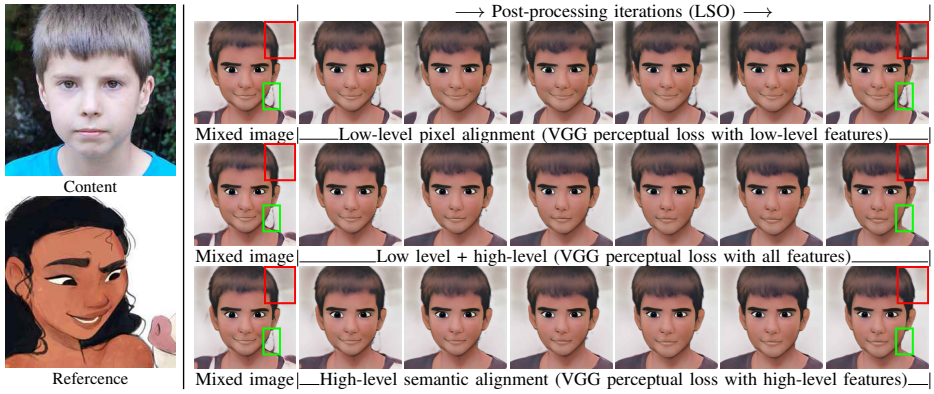


Fig. 14. Using high-level perceptual loss for LSO and aligning images in semantic space. Top: The low-level features of the perceptual loss align images in a pixel-wise manner as MSE does. The low-level perceptual loss cannot fix the artifacts of the mixed images (see the green bounding box) and causes a lot of background noise after LSO (see the red bounding box). Middle: Using all features can remove the artifacts of the mixed image, but low-level features still deteriorate the background. Bottom: High-level features align the mixed image with the content image at the semantic level, remove noise, and do not deteriorate the background.

a) ***HRInversion for Ultra-High Resolution Images****:* To project ultra-high resolution panoramas, we train an Style-INR-GAN using transfer learning on DIV2K dataset [85]. The model is initialized with the source model pre-trained on 1024×1024 FFHQ dataset. Style-INR-GAN can produce images of arbitrary resolution and aspect ratio, and is friendly to inverting non-square images. As shown in Fig. 10, we show two panoramas with different resolutions, proving the effectiveness of HRInversion. For the ultra-high resolution image (Fig. 10c), which contains 6143×1677 pixels, the reconstructed image is impressive (Fig. 10d). Compared with the original image, the color of the reconstructed image is slightly changed (see the color of the trees in Fig. 10c and 10d). We conjecture that the reason is that the latent code $\boldsymbol{w}$ is injected through channels and is responsible for the global color scheme of the image. The spatial noise maps $\boldsymbol{n}$ are responsible for reconstructing the high-frequency details of the image. Therefore, the color of the trees are affected by the global color of the image. We can use this property to do style transfer. As shown in Fig. 11, we combine the latent code $\boldsymbol{w}_s$ of the style image and the noise maps $\boldsymbol{n}_c$ of the content image, so that the reconstructed image $G(\boldsymbol{w}_s, \boldsymbol{n}_c)$ achieves the effect of style transfer.

b) ***HRInversion for 3D-aware GAN inversion****:* We trained a 3D-aware GAN [86] and then use the generator to do 3D-aware GAN inversion. The perceptual loss adopts the HRInversion that uses VGG conv-based perceptual loss and the input is not down-sampled. As shown in Fig. 15, because the generator is 3D aware, we can reconstruct multi-view images for a single real image using GAN inversion. In addition to GAN inversion, as a perceptual loss, HRInversion can be applied to other scenarios such as super-resolution and image restoration. To facilitate people using HRInversion in their own tasks, we provide a minimal implementation of the perceptual loss at our github open source site https://github.com/PeterouZh/HRInversion.

G. Ablation Studies

We perform ablation studies for the perceptual loss and local style optimization (LSO) to understand their individual contributions.

a) ***Layer-wise diagnosis of perceptual loss****:* To understand the properties of the perceptual loss in detail, we analyze the loss layer by layer. Like DGP [15], we adopt the pre-trained discriminator as the perceptual network because it has feature maps of distinct resolution from 1024^2 to 8^2. It is easy for the noise maps $\boldsymbol{n}$ to overfit the input image and impede the analysis for the perceptual loss. Therefore we only optimize the latent code $\boldsymbol{w}$ for this experiment. As shown in Fig. 12, the MSE and the low-level features (from block b1024 to b256 of the discriminator) help restore the low-frequency information of the input image. The reconstructed images are blurry (see the hair). The high-level features (from block b128 to b8) help restore high-frequency details, such as the hair texture. This is a bit counter-intuitive because the higher the layer, the lower the feature resolution. However, it is the low-resolution features that guide the generator to synthesize detailed textures. This indicates that alignment in semantic space (high-level features of the perceptual loss) plays a more important role in reconstructing image details than in pixel space (MSE, low-level features of the perceptual loss).

b) ***Using high-level perceptual loss for LSO****:* We notice that the mixed image and the original image tend to have semantic correspondences rather than pixel correspondences. It inspires us not to use pixel-wise losses such as MSE and low-level perceptual losses when using LSO for post-processing. As shown in Fig. 14 (top), we find that the low-level features of VGG cannot remove the artifacts of the mixed image (see the green bounding box). To make matters worse, the low-level features deteriorates the background (see the red bounding box) because they align images in a pixel-wise manner as the MSE does. Using all features for the perceptual loss does remove artifacts, but the background becomes worse after LSO. Using only high-level features achieves the best results, where LSO aligns images in semantic space and removes noise.

c) ***Local optimization is critical****:* LSO employs two strategies to improve the quality of the mixed images while maintaining the style, namely optimizing only latent codes of early layers and applying L2 regularization to the optimized variables. We perform ablation studies to help understand the

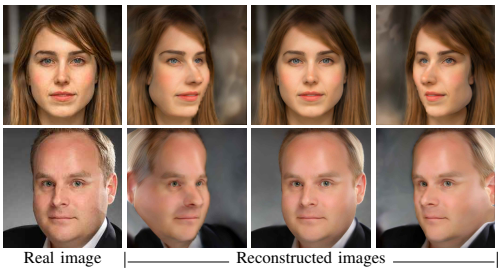


Fig. 15. 3D-aware GAN inversion. We adopt HRInversion as the perceptual loss. Given a single image, we can reconstruct it from different viewpoints.

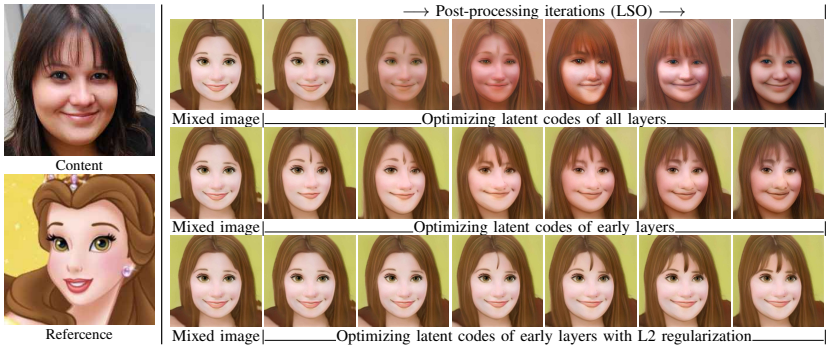


Fig. 16. Local optimization is the key to LSO. Top: Directly optimizing latent codes of all layers degenerates the mixed image into the content image. Middle: Optimizing only latent codes of early layers without L2 regularization retains the color scheme but loses the style (see the eyes). Bottom: Optimizing only latent codes of early layers with L2 regularization achieves a satisfactory result, where it restores the semantics while keeping the style of the mixed image unchanged.

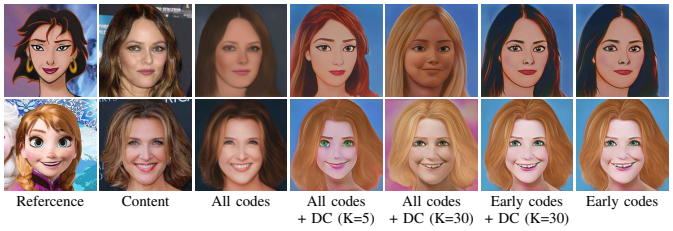


Fig. 17. Effect of direction constraints (DC). K=5 means that only the first five eigenvectors are used as basis vectors. Using too few eigenvectors (K=5) cannot restore semantics, and using too many eigenvectors (K=30) impairs style. Optimizing only latent codes of early layers yields the best results because it exploits the semantically hierarchical nature of the generator. All codes: optimizing latent codes of all layers. Early codes: only optimizing latent codes of early layers.

contributions of each component. As shown in Fig. 16, if we directly optimize the latent codes of all layers, the mixed image will degenerate into the input image. Optimizing only the latent codes of early layers without L2 regularization retains the color scheme of the mixing image but loses the style (see the eyes). Optimizing the latent codes of early layers with L2 regularization aligns the semantics of the mixed image with the input image while retaining the style.

*d) **Optimization Direction Constraints**:* We implement a direction-constrained approach and compare it with our approach that only optimizes the early latent codes. Specifically, we use closed-form factorization [75] to obtain the eigenvectors of the latent space of the mixed generator G_m. Let $\boldsymbol{e}_i \in \mathbb{R}^{512}, i \in \{1, 2, \ldots, 512\}$ denote eigenvectors and $\boldsymbol{e}_1$ is the eigenvector corresponding to the largest eigenvalue. $\boldsymbol{w}$ represents the parameter to be optimized, $\boldsymbol{w} = [w_1, w_2, \ldots, w_K]^T \in \mathbb{R}^K, K \leq 512$. The objective function is given by

$$\min_{\boldsymbol{w},\boldsymbol{n}} d_h \left[G_m(\boldsymbol{w}_m + \sum_{i=1}^{K} w_i \boldsymbol{e}_i + \boldsymbol{\epsilon}, \boldsymbol{n}), \boldsymbol{x}_s \right] + \lambda_{\text{reg}} ||\boldsymbol{w}||_2^2, \quad (9)$$

where $\boldsymbol{w}_m \in \mathcal{W}^+$ is the mixed latent code and keeps fixed. $\boldsymbol{w}$ and $\boldsymbol{n}$ are variables to be optimized. $\boldsymbol{n}$ are the noise variables of StyleGAN2. $\boldsymbol{x}_s$ is the image of the source domain (namely, the content image). $d_h(\cdot)$ is the high-level perceptual loss. λ_{reg} is the coefficient of L2 regularization. In such case, the optimization direction is restricted to the subspace formed by the basis vectors $\{\boldsymbol{e}_i\}_{i=1}^K$.

Fig. 17 presents the results. We deliver several messages. First, directly optimizing latent codes of all layers leads to the stylized images losing their style (third column). Second, using only the first 5 eigenvectors can preserve the style but cannot restore the semantics because the search space is too small (fourth column). Third, using the first 30 eigenvectors restores semantics but impairs style (column 5). In contrast, only optimizing codes of early layers restores semantics without compromising style (column 6 and 7). Furthermore, imposing direction constraints does not significantly improve the results of only optimizing latent codes of early layers (column 6 *vs.* 7). These results indicate that optimizing only latent codes of early layers is important to preserve style while recovering semantics.

V. Conclusion

This paper diagnoses perceptual losses comprehensively on the GAN inversion task. We find that the input resolution of the perceptual loss is important and propose to use the features of convolutional layers to compute the perceptual loss to attenuate the effect of scale. We apply HRInversion to a cross-domain image synthesis task and propose a post-processing approach, LSO, to improve the initially unsatisfactory stylized images. Experiments validate that our approach is capable of synthesizing images of various styles. Our approach reveals that different layers of the perceptual loss play different roles. It is necessary to adjust the perceptual loss according to different tasks. We also apply HRInversion to other tasks such as non-square image GAN inversion, style transfer, and 3D-aware GAN inversion, demonstrating its wide range of applications.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *NeurIPS*, 2014. 1

[2] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *CVPR*, 2019. 1, 3, 7

[3] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in *CVPR*, 2020. 1, 3, 4, 7, 8

[4] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training Generative Adversarial Networks with Limited Data," in *NeurIPS*, 2020. 1, 4, 7

[5] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-Free Generative Adversarial Networks," in *NeurIPS*, 2021. 1

[6] Q. Duan, L. Zhang, and X. Gao, "Simultaneous Face Completion and Frontalization via Mask Guided Two-Stage GAN," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, 2022. 1

[7] L. Zhang, H. Yang, T. Qiu, and L. Li, "AP-GAN: Improving Attribute Preservation in Video Face Swapping," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, 2022. 1

[8] W. Yan, Y. Zeng, and H. Hu, "Domain Adversarial Disentanglement Network With Cross-Domain Synthesis for Generalized Face Anti-Spoofing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, 2022. 1

[9] Y. Jun, C. Jiang, R. Li, C.-W. Luo, and Z.-F. Wang, "Real-Time 3-D Facial Animation: From Appearance to Internal Articulators," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 4, 2018. 1

[10] Q. Li, X. Wang, B. Ma, X. Wang, C. Wang, S. Gao, and Y. Shi, "Concealed Attack for Robust Watermarking Based on Generative Model and Perceptual Loss," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, 2022. 1

[11] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?" in *ICCV*, 2019. 1, 2, 3, 4, 5, 8

[12] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative Visual Manipulation on the Natural Image Manifold," in *ECCV*, 2018. 1, 3

[13] T. M. Dinh, A. T. Tran, R. Nguyen, and B.-S. Hua, "HyperInverter: Improving StyleGAN Inversion via Hypernetwork," in *CVPR*, 2022. 1

[14] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models," in *CVPR*, 2020. 1, 3

[15] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation," in *ECCV*, 2020. 1, 3, 11

[16] K. C. K. Chan, X. Wang, X. Xu, J. Gu, and C. C. Loy, "GLEAN: Generative Latent Bank for Large-Factor Image Super-Resolution," in *CVPR*, 2021. 1, 3

[17] J. Gu, Y. Shen, and B. Zhou, "Image Processing Using Multi-Code GAN Prior," in *CVPR*, 2020. 1

[18] 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. 1, 3, 8, 9

[19] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture Synthesis Using Convolutional Neural Networks," in *NeurIPS*, 2015. 1, 3

[20] ——, "A Neural Algorithm of Artistic Style," *arXiv:1508.06576 [cs, q-bio]*, 2015. 1, 3

[21] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *ECCV*, 2016. 1, 3, 4

[22] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-Resolution with Deep Convolutional Sufficient Statistics," *arXiv:1511.05666 [cs]*, 2016. 1

[23] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic Image Synthesis with Spatially-Adaptive Normalization," in *CVPR*, 2019. 1

[24] X. Liu, G. Yin, J. Shao, X. Wang, and H. Li, "Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis," in *NeurIPS*, 2019. 1

[25] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to Edit the Embedded Images?" in *CVPR*, 2020. 2, 3, 4, 8

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009. 2, 4

[27] S. Laine, "Feature-Based Metrics for Exploring the Latent Space of Generative Models," in *ICLRW*, 2018. 2, 3, 4, 8

[28] J. N. M. Pinkney and D. Adler, "Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains," *arXiv:2010.05334 [cs]*, 2020. 2, 3, 5, 7, 8, 10

[29] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks," in *ICLR*, 2019. 2, 3, 5

[30] C. Yang, Y. Shen, and B. Zhou, "Semantic Hierarchy Emerges in Deep Generative Representations for Scene Synthesis," *International Journal of Computer Vision*, 2020. 2, 5

[31] M. Liu, Q. Li, Z. Qin, G. Zhang, P. Wan, and W. Zheng, "BlendGAN: Implicitly GAN Blending for Arbitrary Stylized Face Generation," in *NeurIPS*, 2021. 2

[32] Y. Men, Y. Yao, M. Cui, Z. Lian, X. Xie, and X.-S. Hua, "Unpaired Cartoon Image Synthesis via Gated Cycle Mapping," in *CVPR*, 2022. 2

[33] A. Dosovitskiy and T. Brox, "Generating Images with Perceptual Similarity Metrics based on Deep Networks," in *NeurIPS*, 2016. 3

[34] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," in *ICCV*, 2017. 3

[35] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *CVPR*, 2016. 3

[36] K. Liao, C. Lin, Y. Zhao, and M. Gabbouj, "DR-GAN: Automatic Radial Distortion Rectification Using Conditional GAN in Real-Time," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, 2020. 3

[37] P. Wang, H. Zhu, H. Huang, H. Zhang, and N. Wang, "TMS-GAN: A Twofold Multi-Scale Generative Adversarial Network for Single Image Dehazing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, 2022. 3

[38] S. Xu, D. Liu, and Z. Xiong, "E2I: Generative Inpainting From Edge to Image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, 2021. 3

[39] F. Peng, L. Yin, and M. Long, "BDC-GAN: Bidirectional Conversion between Computer-generated and Natural Facial Images for Anti-forensics," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 3

[40] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "GAN Inversion: A Survey," *arXiv:2101.05278 [cs]*, 2021. 3

[41] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, "Swapping Autoencoder for Deep Image Manipulation," in *NeurIPS*, 2020. 3

[42] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-Domain GAN Inversion for Real Image Editing," in *ECCV*, 2020. 3

[43] Y. Alaluf, O. Patashnik, and D. Cohen-Or, "ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement," in *ICCV*, 2021. 3

[44] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the Latent Space of GANs for Semantic Face Editing," in *CVPR*, 2020. 3

[45] Y. Xu, Y. Shen, J. Zhu, C. Yang, and B. Zhou, "Generative Hierarchical Features from Synthesizing Images," in *CVPR*, 2021. 3

[46] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation," in *CVPR*, 2021. 3

[47] Y.-D. Lu, H.-Y. Lee, H.-Y. Tseng, and M.-H. Yang, "Unsupervised Discovery of DisentangledManifolds in GANs," *arXiv:2011.11842 [cs]*, 2020. 3

[48] T. Wang, Y. Zhang, Y. Fan, J. Wang, and Q. Chen, "High-Fidelity GAN Inversion for Image Attribute Editing," in *CVPR*, 2022. 3

[49] A. Raj, Y. Li, and Y. Bresler, "GAN-based Projector for Faster Recovery with Convergence Guarantees in Linear Inverse Problems," in *ICCV*, 2019. 3

[50] E. Collins, R. Bala, B. Price, and S. Süsstrunk, "Editing in Style: Uncovering the Local Semantics of GANs," in *CVPR*, 2020. 3

[51] G. Daras, A. Odena, H. Zhang, and A. G. Dimakis, "Your Local GAN: Designing Two Dimensional Local Attention Mechanisms for Generative Models," in *CVPR*, 2020. 3

[52] D. Bau, S. Liu, T. Wang, J.-Y. Zhu, and A. Torralba, "Rewriting a Deep Generative Model," in *ECCV*, 2020. 3

[53] M. Huh, R. Zhang, J.-Y. Zhu, S. Paris, and A. Hertzmann, "Transforming and Projecting Images into Class-conditional Generative Networks," in *ECCV*, 2020. 3

[54] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "StyleGAN2 Distillation for Feed-forward Image Manipulation," in *ECCV*, 2020. 3

[55] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," in *ICLR*, 2020. 3

[56] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering Interpretable GAN Controls," in *NeurIPS*, 2020. 3

[57] Z. Wu, D. Lischinski, and E. Shechtman, "StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation," in *CVPR*, 2021. 3

[58] A. Cherepkov, A. Voynov, and A. Babenko, "Navigating the GAN Parameter Space for Semantic Image Editing," in *CVPR*, 2021. 3

[59] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *CVPR*, 2017. 3

[60] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," in *CVPR*, 2018. 3

[61] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *ICCV*, 2017. 3, 7

[62] 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. 3

[63] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse Image Synthesis for Multiple Domains," in *CVPR*, 2020. 3, 8, 10

[64] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal Unsupervised Image-to-Image Translation," *arXiv:1804.04732 [cs, stat]*, 2018. 3

[65] S. Yang, L. Jiang, Z. Liu, and C. C. Loy, "Pastiche Master: Exemplar-Based High-Resolution Portrait Style Transfer," in *CVPR*, 2022. 3, 7, 8, 10

[66] D. Lee, J. Y. Lee, D. Kim, J. Choi, and J. Kim, "Fix the Noise: Disentangling Source Feature for Transfer Learning of StyleGAN," in *arXiv:2204.14079 [Cs]*, no. arXiv:2204.14079, 2022. 3

[67] J. Huang, J. Liao, and S. Kwong, "Unsupervised Image-to-Image Translation via Pre-trained StyleGAN2 Network," *arXiv:2010.05713 [cs]*, 2020. 3, 7

[68] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," in *CVPR*, 2019. 4

[69] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space," in *CVPR*, 2019. 4

[70] Z. Chen and H. Zhang, "Learning Implicit Fields for Generative Shape Modeling," in *CVPR*, 2019. 4

[71] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," in *CVPR*, 2018. 4

[72] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 4

[73] R. Wightman, "PyTorch image models," *GitHub repository*, 2019. 4, 7

[74] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv:2012.12877 [cs]*, 2020. 6, 8

[75] Y. Shen and B. Zhou, "Closed-Form Factorization of Latent Semantics in GANs," in *CVPR*, 2021. 5, 12

[76] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping," in *CVPR*. arXiv, 2020. 6

[77] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in *ICLR*, 2018. 6, 9

[78] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *CVPR*, 2019. 6

[79] J. Huo, W. Li, Y. Shi, Y. Gao, and H. Yin, "WebCaricature: A benchmark for caricature recognition," in *British Machine Vision Conference*, 2018. 8, 10

[80] J. Huo, Y. Gao, Y. Shi, and H. Yin, "Variation robust cross-modal metric learning for caricature recognition," in *ACM Multimedia Workshop*, 2017. 8, 10

[81] G. Branwen, Anonymous, and D. Community, "Danbooru2019 portraits: A large-scale anime head illustration dataset," 2019. 8, 10

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 8

[83] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," in *ECCV*, 2018. 8

[84] Y. Liu, H. Chen, Y. Chen, W. Yin, and C. Shen, "Generic Perceptual Loss for Modeling Structured Output Dependencies," in *CVPR*, 2021. 8

[85] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *CVPR Workshops*, 2017. 11

[86] P. Zhou, L. Xie, B. Ni, and Q. Tian, "CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis," *arXiv:2110.09788 [cs, eess]*, 2021. 11

Peng Zhou received his B.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016. He has been working towards a Ph.D. at the Department of Electronic Engineering, Shanghai Jiao Tong University, since 2016. His research interests include image generation, avatar creation and animation.

Lingxi Xie is currently a senior researcher at Cloud BU, Huawei Inc. He received his B.E. and Ph.D. in engineering, both from Tsinghua University, in 2010 and 2015, respectively. He also served as a postdoctoral researcher at the CCVL lab from 2015 to 2019, having moved from the University of California, Los Angeles, to Johns Hopkins University. Lingxi's research interests are in computer vision, particularly the application of deep learning models. His research covers image classification, object detection, semantic segmentation, and other vision tasks. He is also interested in medical image analysis, especially object segmentation in CT and MRI scans.

Bingbing Ni received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree from the National University of Singapore, Singapore, in 2011. He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. Before that, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. He was with Microsoft Research Asia, Beijing, China, as a Research Intern in 2009. He was also a Software Engineer Intern with Google Inc., Mountain View, CA, USA, in 2010. Dr. Ni was a recipient of the Best Paper Award from PCM11 and the Best Student Paper Award from PREMIA08.

Lin Liu Lin Liu received the B.S. degree from University of Science and Technology of China, in 2019. He is currently pursuing the PhD degree with University of Science and Technology of China. His research interests include computer vision and machine learning. Now he is a student research intern at Huawei working on low level vision tasks.

Qi Tian is currently the Chief Scientist at Cloud BU, Huawei Inc. He was a tenured associate professor during 2008-2012 and a tenure-track assistant professor during 2002-2008. From 2008 to 2009, he took faculty leave for one year at Microsoft Research Asia (MSRA) as Lead Researcher in the Media Computing Group. Dr. Tian received his Ph.D. in ECE from the University of Illinois at Urbana-Champaign (UIUC) in 2002 and received his B.E. degree in electronic engineering from Tsinghua University in 1992 and his M.S. degree in ECE from Drexel University in 1996. Dr. Tian's research interests include multimedia information retrieval, computer vision, pattern recognition. He has published over 360 refereed journal and conference papers. He was the coauthor of a Best Paper at ACM ICMR 2015, a Best Paper at PCM 2013, a Best Paper at MMM 2013, a Best Paper at ACM ICIMCS 2012, a Top 10% Paper at MMSP 2011, a Best Student Paper at ICASSP 2006, a Best Student Paper Candidate at ICME 2015, and a Best Paper Candidate at PCM 2007. Dr. Tian received the 2017 UTSA President's Distinguished Award for Research Achievement; the 2016 UTSA Innovation Award; the 2014 Research Achievement Awards from the College of Science, UTSA; the 2010 Google Faculty Award; and the 2010 ACM Service Award. He is an associate editor of many journals and on the Editorial Board of the Journal of Multimedia (JMM) and Journal of Machine Vision and Applications (MVA). He is a fellow of the IEEE.