A Style-Based GAN Encoder for High Fidelity Reconstruction of Images and Videos

Xu Yao[1], Alasdair Newson[1], Yann Gousseau[1], and Pierre Hellier[2]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2] InterDigital R&I, France
yaoxu.fdu@gmail.com

Abstract. We present a new encoder architecture for the inversion of Generative Adversarial Networks (GAN). The task is to reconstruct a real image from the latent space of a pre-trained GAN. Unlike previous encoder-based methods which predict only a latent code from a real image, the proposed encoder maps the given image to both a latent code and a feature tensor, simultaneously. The feature tensor is key for accurate inversion, which helps to obtain better perceptual quality and lower reconstruction error. We conduct extensive experiments for several style-based generators pre-trained on different data domains. Our method is the first feed-forward encoder to include the feature tensor in the inversion, outperforming the state-of-the-art encoder-based methods for GAN inversion. Additionally, experiments on video inversion show that our method yields a more accurate and stable inversion for videos. This offers the possibility to perform real-time editing in videos. *Code is available at `https://github.com/InterDigitalInc/FeatureStyleEncoder`.*

Keywords: GAN inversion, styleGAN encoder, latent space

1 Introduction

The image synthesis power of Generative Adversarial Networks [12] (GAN) has been amply demonstrated by a great quantity of work on such architectures. However, since a GAN only decodes an image from a probabilistic latent space, a significant research problem is how to *encode* images into the latent space of a pretrained GAN, especially in the case of real (photographic) images, as opposed to *synthetic* images, which are generated by sampling in the latent space. Recent studies [34,17,44,35] have shown that it is possible to control semantic attributes of synthetic images by manipulating the latent space of a pre-trained GAN. However, an efficient encoding method, necessary for real images, still remains an open problem especially in the case of these editing tasks.

Among the many studies on GAN inversion, recent works have been primarily focused on style-based generators [22,23,21], because of their excellent performance in high quality image synthesis. Unlike traditional generative models which feed the latent code though the input layer only, style-based generators feed latent code to each scale of the generator to control the style of the generated

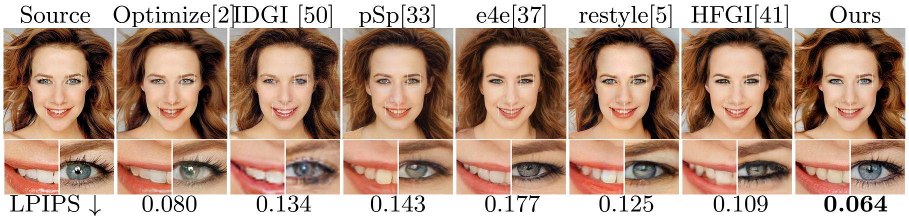


Fig. 1. Inversion of a real image in the latent space of StyleGAN2. We compare our model against state-of-the-art for the inversion of StyleGAN2 [23] pre-trained on face domain. Our method outperforms the state-of-the-art by up to 20% − 50% in LPIPS distance[49].

image. This multi-scale generation is one of the main strengths of style-based generators. It is clear that the architecture's success is based on the separation of the latent code, which is a vector that acts *globally* at each scale, from the feature tensor, which has a *spatial* organisation. Exploiting this separation is a key component of the encoder we propose here.

To invert a pretrained GAN, the current solutions can be divided into two types: optimization-based methods and encoder-based methods. Optimization-based methods use some form of gradient descent in the latent space to find the code that minimizes a reconstruction error. Encoder-based methods, on the other hand, train a neural network that projects from the image space to the latent space. While the optimization is straight-forward to perform, it has significant shortcomings. Firstly, the inverted latent codes do not necessarily lie on the original latent space, since the optimization is carried out locally with respect to one image, making them difficult to use for editing tasks, as shown by [37,5]. Secondly, the approach is computationally expensive, requiring an optimization for each image. The encoder-based approach is much faster and the inverted latent codes are more regularized and *better suited for editing*.

Currently, all the encoder-based GAN inversion methods use the latent code only, ignoring the above-mentioned feature tensors. Unfortunately, while the inversion results are globally perceptually similar to the input, they lack crisp finer spatial details and appear over-smoothed. This is coherent, since the latent codes act *globally*, thus spatially localized details are difficult to preserve. Several optimization-based approaches have identified this problem, and show that considering the feature tensors leads to results of better perceptual quality. However, they present all the drawbacks of optimization-based approaches mentioned above (slow, unconstrained latent code).

To have the best of both worlds, we consider an encoder-based approach which modifies both the latent code and the feature tensors simultaneously, an approach that currently does not exist in the literature. We learn an encoder which maps an image to a feature tensor and a latent code, simultaneously. This design significantly improves the perceptual quality of the inversion and achieves a balanced trade-off between reconstruction quality and editing capacity. The main contributions of our paper can be summarized as follows:

- We propose a new GAN encoder architecture, which is the first feed-forward encoder to include the feature tensor in the inversion. To train the encoder, we present a new training process, which learns two inversions simultaneously, on both real and synthetic images, which significantly improves the perceptual quality;
- We present a novel latent space editing approach, which allows us to leverage existing editing methods for style-based generators. This way, we achieve editing results that are comparable to state-of-the-art methods while preserving the high fidelity inversion;
- We conduct extensive experiments to show that our model greatly outperforms state-of-the-art methods on inversion and editing tasks on images and videos. In particular, we improve the perceptual metrics by a very large margin (50%). In addition, we show that the video inversion results of our method is more consistent and stable, which favors further editing on videos.

2 Related works

The goal of our work is to learn an encoder for projecting real images to the latent space of a pre-trained GAN. Much of the recent literature on GAN inversion pays particular attention to style-based generators [22,23,20,21], as their latent spaces are better disentangled and have improved editing properties.

Style-based Generator. Karras *et al*. proposed the first style-based generator, named StyleGAN [22]. Unlike traditional generative models which feed the latent code though the input layer only, a style-based generator feeds latent code through adaptive instance normalization at each convolution layer to control the style of the generated image. The perceptual quality and variety of the StyleGAN synthetic images surpassed previous image generative models [19,7]. In StyleGAN2 [23], the image quality was improved further by introducing weight demodulation and path length regularization and redesigning the generator normalization. The StyleGAN2-Ada [20] explored the possibility to train a GAN model with limited data regimes, by using an adaptive discriminator augmentation mechanism that significantly stabilizes training. The third generation, alias-free GAN [21], addressed the aliasing artifacts in the generator, by employing small architectural changes to discard unwanted information and boost the generator to be fully equivariant to translation and rotation.

Latent Space Editing. The motivation of GAN inversion is to achieve real image editing using the latent space of a pretrained GAN model. Various studies show it possible to edit synthetic images by manipulating the corresponding latent code. Local semantic editing can be achieved by optimizing the latent code directly [2,27]. To explore high level semantic information in the latent space, learning based techniques have been proposed. These techniques include unsupervised exploration [39], learning linear SVM models [34], principle component analysis on the latent codes [17], and k-means clustering of the activation features [10]. To achieve better disentangled editing, [3,30,47,36,14] proposed to learn neural networks in the latent space. The recent works [35,40] discovered interpretable transforms by directly decomposing the weights or feature maps of pre-trained GANs. Additionally, [6,26] modify the style-based GAN architecture and retrain it for better disentanglement in image generation. [31,29,24] train jointly an encoder and a style-based decoder architecture for image manipulations.

GAN Inversion. The goal of GAN inversion is to encode a real image to the latent space of a pretrained GAN, so that the image generated from the inverted latent code is the reconstruction of the input image. Among the rich literature on GAN inversion [45], the approaches addressing style-based generators can be classified into two main types: optimization-based methods [1,16,46,51,18] and encoder-based models [33,37,5,41,43]. There are also hybrid methods [50,8,48] which mix the two previous ones. The optimization-based methods produce the inverted latent code by minimizing the reconstruction error on the input image. For StyleGAN inversion, Abdal *et al.* [1] proposed to embed the input image in an extended latent space $\mathcal{W}^+$, which offers greater flexibility and improves the reconstruction quality. Recent works show that including a feature tensor in the optimization helps preserve more spatial details and improves the perceptual quality [51,18]. Despite the satisfying reconstruction quality, optimization-based methods usually present lower editing quality due to the random element of the optimization process. To better regularize/stabilize the inversion, encoder-based methods train an encoder to map real images to the latent space of the pretrained generator. Richardson *et al.* [33] proposed the first baseline to learn an encoder for StyleGAN inversion. To improve editing capacity, Tov *et al.* [37] proposed a regularization term which forces the inverted latent code in $\mathcal{W}^+$ to lie closer to the original latent space. A recent concurrent work of Wang *et al.* [41] formulated the inversion task to a data compression problem and proposed an adaptive distortion alignment module to improve the reconstruction quality. On the other hand, hybrid methods take the inverted latent code from a pretrained encoder as initialization and perform optimization on it. Zhu *et al.* [50] proposed to learn a domain-guided encoder and use it as a regularizer for domain-regularized optimization. However, despite the gain in the reconstruction quality, the optimization step makes hybrid methods less suited for video inversion and editing.

3 Method

3.1 Overview

A style-based generator, such as StyleGAN [22,23,21], consists of a mapping network and a generator $\mathbf{G}$. The mapping network first maps a random latent code $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^{512}$ to an intermediate latent code $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^{512}$, which is further used to scale and bias the feature tensors. We denote a feature tensor (also called feature map) with $\mathbf{F} \in \mathcal{F} \subset \mathbb{R}^{h \times w \times c}$. The parameters (h, w, c) correspond to the spatial size and the number of channels of the tensor. Thus, contrary to the latent codes, the feature tensors have a 2D spatial organisation. Each feature tensor is the output of an upsampling from a lower resolution, followed by an

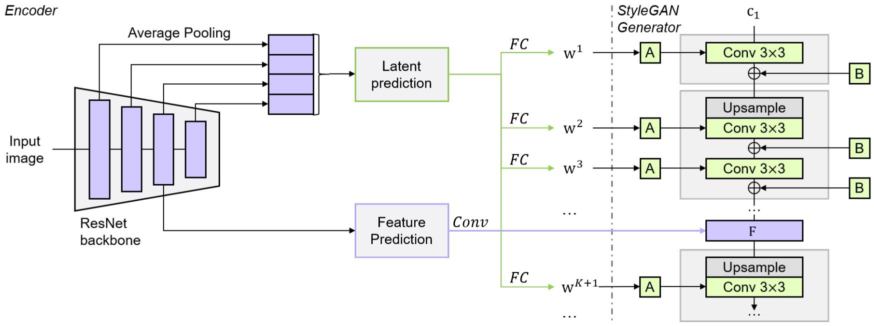


Fig. 2. Encoder Architecture. Our model consists of a ResNet backbone and two output branches: one for latent code prediction, the other for feature tensor prediction. The inverted latent code $\mathbf{w}$ is a concatenation of N latent blocks $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^N\}$, each controlling a separate convolution layer in the generator. During generation, we replace the feature tensors at the K^{th} convolution layer of the generator with the encoded feature tensor $\mathbf{F}$, and synthesize the inversion with the latent blocks $\{\mathbf{w}^{K+1}, ..., \mathbf{w}^N\}$. K is a fixed parameter, chosen so that reconstruction is accurate and editing can be performed efficiently

AdaIn layer controlled by the latent codes, and finally a convolution. At the coarsest layer, the input is a constant feature tensor, which is learned during training. See Figure 2 for an illustration. In this Figure, we can see the clear separation between latent codes and feature tensors, which is so important to StyleGAN's success.

To project a *synthetic* image $\mathbf{G}(\mathbf{w})$ to the latent space, it is possible to compute the latent code in the original latent space $\mathcal{W}$ and achieve a satisfying inversion. However, it is much more difficult to project a real image to the original latent space [23], due to the gap between the real data distribution and the synthetic one. An alternative is to project real images to an extended latent space $\mathcal{W}^+$ [1], where $\mathbf{w} \in \mathcal{W}^+$ is a concatenation of N latent blocks $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^N\}$, each controlling a feature tensor in the generator.

Current encoder based methods for GAN inversion learn only the latent codes. Their inversion results are globally perceptually similar to the input but fail to capture finer details. Therefore, it is preferable to include a learned feature tensor to preserve these spatial details. The optimization-based methods [51,18] show that including these feature tensors in the optimization process help to preserve spatial details. Performing optimization on both the latent code and feature tensors yields near perfect reconstruction on real images.

In our work, we aim to have the best of both worlds: we wish to achieve this high reconstruction fidelity, while maintaining the speed and editing capacity of an encoder. Thus, we propose an encoder architecture which projects an image to a latent code $\mathbf{w} \in \mathcal{W}^+$, and a feature tensor $\mathbf{F} \in \mathcal{F}$. This feature tensor is chosen at a fixed layer K of the generator.

3.2 Encoder Architecture

The basic structure of our encoder is modelled on the classic approach used by most previous works on style-based GAN inversion [33,37,5,43,41], which employ a ResNet backbone. Different from the existing methods, as shown in Figure 2, we have two output branches:

- A latent prediction branch to encode the latent code $\mathbf{w} \in \mathcal{W}^{+}$. The ResNet backbone contains four blocks, each down-sampling the input tensors by a factor of 2. Given an input image, we extract the tensors after each block. Then the four groups of tensors are passed through an average pooling layer, concatenated and flattened to produce the latent prediction. This is then mapped to the latent code $\mathbf{w} = \{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^N\}$. Each latent block $\mathbf{w}^i$ is generated from a different mapping network, expressed by a single fully connected layer.
- A feature prediction branch, where the ResNet tensors, extracted after the penultimate block of the ResNet backbone, are passed through a convolutional network to encode the feature tensor $\mathbf{F} \in \mathcal{F}$ (see Figure 2). This network is implemented with two convolutional layers, with batch normalization in between. Please note that the ResNet tensors are not the same as those of StyleGAN. We refer to the StyleGAN tensors (which control the generation) as *feature tensors*. Let $\mathbf{G}(\mathbf{w}^{1:K})$ denote the feature tensors at the K^{th} convolution layer of the generator. We replace $\mathbf{G}(\mathbf{w}^{1:K})$ with the encoded feature tensor $\mathbf{F}$, and use the rest of the latent codes $\{\mathbf{w}^{K+1}, ..., \mathbf{w}^N\}$ to generate the inversion $\mathbf{G}(\mathbf{F}, \mathbf{w}^{K+1:N})$. We choose $K = 5$ for a balance between the inversion quality and editing capacity, leading to $\mathcal{F} \subset \mathbb{R}^{16\times16\times512}$.

To summarize the entire process, our encoder produces the K^{th} feature tensor of the StyleGAN generator, simultaneously with all the latent codes. Due to the sequential nature of StyleGAN, $\mathbf{G}(\mathbf{w}^{1:K})$ and $\{\mathbf{w}^{K+1}, ..., \mathbf{w}^N\}$ completely determine the output image. Another way of seeing this is that we have “started” the generation from layer K, ignoring the previous layers. Note that even if the latent codes of previous layers $\{\mathbf{w}^1, ..., \mathbf{w}^K\}$ are not used for the inversion, they will be used later for *editing*. The choice of K is crucial to achieve a balance between good reconstruction and style editing. We have studied this choice carefully, and show results for different values in the Supplementary Material.

3.3 Editing

In a style-based generator, the styles corresponding to coarse layers control high-level semantic attributes, the styles of the middle layers control smaller scale features, and the last layers control micro structures. Given a latent code $\mathbf{w}$, let us consider that we have a modified latent code $\tilde{\mathbf{w}} = \mathbf{w} + \Delta\mathbf{w}$ corresponding to a desired editing, obtained from a latent space editing method [34,17,35].

Contrary to the case of inversion, we now wish to modify the latent codes of *all* layers, to achieve editing. For this, we start by obtaining the initial inversion of the input image, which gives us the latent code $\mathbf{w}$ and the feature tensor $\mathbf{F}$. Recall

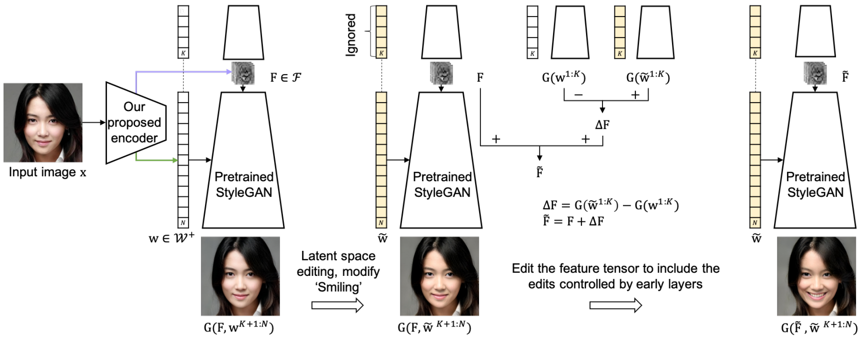


Fig. 3. Editing process. To edit an input image $\mathbf{x}$, we first encode it to the latent code $\mathbf{w}$ and the feature tensor $\mathbf{F}$. We then use a latent space editing method to transform $\mathbf{w}$ into $\tilde{\mathbf{w}}$, which corresponding to a desired manipulation. If we use $\tilde{\mathbf{w}}$ and $\mathbf{F}$ to generate the output image $\mathbf{G}(\mathbf{F}, \tilde{\mathbf{w}}^{K+1:N})$, the changes corresponding to early layers will be ignored. Therefore, we need to modify $\mathbf{F}$ to include these edits. We do this by extracting the feature tensors at the K^{th} layer $\mathbf{G}(\mathbf{w}^{1:K})$ and $\mathbf{G}(\tilde{\mathbf{w}}^{1:K})$, computing the difference and adding it to $\mathbf{F}$ to obtain the modified feature tensor $\tilde{\mathbf{F}}$. Finally, we use this new feature tensor $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{w}}$ to generate the edited image $\mathbf{G}(\tilde{\mathbf{F}}, \tilde{\mathbf{w}}^{K+1:N})$

that the inversion is determined by the feature tensor $\mathbf{F}$ and the latent codes $\{\mathbf{w}^{K+1}, ..., \mathbf{w}^{N}\}$. Thus, the changes corresponding to early layers $\{\mathbf{w}^{1}, ..., \mathbf{w}^{K}\}$ will be ignored. To preserve these editing effects, we need to modify $\mathbf{F}$. As shown in Figure 3, we do this by extracting the feature tensors at the K^{th} layer $\mathbf{G}(\mathbf{w}^{1:K})$ and $\mathbf{G}(\tilde{\mathbf{w}}^{1:K})$. We suppose that the difference between them should be close to that between $\mathbf{F}$ and the modified feature tensor $\tilde{\mathbf{F}}$. Therefore, we can find $\tilde{\mathbf{F}}$ as follows:

$$\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{G}(\tilde{\mathbf{w}}^{1:K}) - \mathbf{G}(\mathbf{w}^{1:K}). \tag{1}$$

Finally, we use this new feature tensor $\tilde{\mathbf{F}}$ and the rest of the modified latent codes $\{\tilde{\mathbf{w}}^{K+1}, ..., \tilde{\mathbf{w}}^{N}\}$ to generate the edited image $\mathbf{G}(\tilde{\mathbf{F}}, \tilde{\mathbf{w}}^{K+1:N})$.

4 Training

Previous methods on GAN inversion [33,37,5,41,43] use only real images as training data. However, the perceptual quality of their inversion results is not as good as the synthetic images generated by StyleGAN. An intuitive explanation is that there is a difference between the data distributions of the real and synthetic images. Recall that the encoder project a given image to the extended latent space $\mathcal{W}^+$ while synthetic images can be reconstructed from the original latent space $\mathcal{W}$. If synthetic images are not viewed by the encoder, the resulting latent codes may not perform in the same way as those of the original latent space. Therefore, we use both synthetic and real images as training data.

4.1 Losses

As mentioned above, the proposed encoder inverts an input image $\mathbf{x}$ to a latent code $\mathbf{w}$, and a feature tensor $\mathbf{F}$. To ensure the editing capacity of the latent code, the encoder is trained on two inversions simultaneously - one generated with only the latent code $\tilde{\mathbf{x}}_1 = \mathbf{G}(\mathbf{w})$ and the other generated with both the feature tensor and the latent code $\tilde{\mathbf{x}}_2 = \mathbf{G}(\mathbf{F}, \mathbf{w}^{K+1:N})$.

Pixel-wise reconstruction loss In the case of a synthetic image, the reconstruction is measured using mean squared error (MSE) on $\tilde{\mathbf{x}}_1$ only. In this special case, the ground-truth latent code exists so theoretically a perfect inversion can be obtained. For a real image, the ground-truth latent code may not exist, so a per-pixel constraint may be too restrictive. The loss is expressed as:

$$\mathcal{L}_{mse} = ||\mathbf{G}(\mathbf{w}) - \mathbf{x}||_2. \tag{2}$$

Multi-scale perceptual loss A common problem of the previous methods is the lack of sharpness of the inversion results, despite using the per-pixel MSE. To tackle this, we propose a multi-scale loss design which enables the reconstruction of finer details. Given an input image $\mathbf{x}$ and its inversion $\tilde{\mathbf{x}}$, the multi-scale perceptual loss is defined as:

$$\mathcal{L}_{m\_lpips}(\tilde{\mathbf{x}}) = \sum_{i=0}^{2} ||\mathbf{V}(\lfloor \tilde{\mathbf{x}} \rfloor_i) - \mathbf{V}(\lfloor \mathbf{x} \rfloor_i)||_2, \tag{3}$$

where $\lfloor . \rfloor_i$ refers to downsampling by a scale factor 2^i and $\mathbf{V}$ denotes the feature extractor. This design allows the encoder to capture the perceptual similarities at different scales, which favors the perceptual quality of the inversion. This loss is applied on both inversions.

Feature reconstruction To ensure the possibility of using Eq. (1) to edit the encoded feature tensor $\mathbf{F}$, $\mathbf{F}$ should be similar to the feature tensors at the K^{th} convolution layer of the generator, denoted by $\mathbf{G}(\mathbf{w}^{1:K})$. Therefore, we propose a feature reconstruction loss, which favors the encoded feature tensor to stay close to the original feature space. This term is defined as:

$$\mathcal{L}_{f\_recon} = ||\mathbf{F} - \mathbf{G}(\mathbf{w}^{1:K})||_2. \tag{4}$$

The total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{mse} + \lambda_1 \mathcal{L}_{m\_lpips} + \lambda_2 \mathcal{L}_{f\_recon}, \tag{5}$$

where $\lambda_1 = 0.2$ and $\lambda_2 = 0.01$ are weights balancing each loss.

Face Inversion For the inversion of a styleGAN model pre-trained on face domain, we adopt the multi-layer identity loss and the face parsing loss introduced by [43]. Given an input image $\mathbf{x}$ and its inversion $\tilde{\mathbf{x}}$, the multi-layer identity loss is defined as:

$$\mathcal{L}_{id}(\tilde{\mathbf{x}}) = \sum_{i=1}^{5} (1 - \langle \mathbf{R}_i(\tilde{\mathbf{x}}), \mathbf{R}_i(\mathbf{x}) \rangle), \tag{6}$$

where $\mathbf{R}$ is the pre-trained ArcFace network [11]. The face parsing loss is defined as:

$$\mathcal{L}_{parse}(\tilde{\mathbf{x}}) = \sum_{i=1}^{5}(1 - \langle \mathbf{P}_i(\tilde{\mathbf{x}}), \mathbf{P}_i(\mathbf{x})\rangle), \tag{7}$$

where $\mathbf{P}$ is a pre-trained face parsing model [52]. These two above-mentioned losses are applied on both inversions. Hence the total loss for face inversion is:

$$\mathcal{L}_{face} = \mathcal{L}_{total} + \lambda_3 \mathcal{L}_{id} + \lambda_4 \mathcal{L}_{parsing}, \tag{8}$$

where $\lambda_3 = 0.1$ and $\lambda_4 = 0.1$ are weights balancing the identity preserving and face parsing terms.

4.2 Implementation details

We train the proposed encoder for the inversion of several style-based generators pre-trained on various domains, specifically, for StyleGAN2 [23] on faces and cars, and StyleGAN2-Ada [20] on cats and dogs. In addition, we show preliminary results for the inversion of StyleGAN3 [21] on faces. For each generator pre-trained on a specific domain, a separate encoder is trained. During the training, we use a batch size of 4, each batch containing two real images and two synthetic images. The model is trained for 12 epochs, using $10K$ iterations per epoch. The learning rate is 10^{-4} for the first 10 epochs and is divided by ten for the last 2 epochs. For the face domain, we minimize Eq. (8), using FFHQ [22] for training, and CelebA-HQ [19] for evaluation. For the car domain, we minimize Eq. (5), using Stanford Cars [25] training set for training, and the corresponding test set for evaluation. For the cat/dog domain, we minimize Eq. (5), using AFHQ Cats/Dogs [9] train set for training, and the corresponding test set for evaluation.

5 Experiments

In this section, we compare our method with the current state-of-the-art GAN inversion methods. We conduct the evaluation from two aspects: inversion quality and editing capacity. We also show results on videos as well as ablation studies.

5.1 Inversion

We evaluate our model against the current state-of-the-art encoder-based GAN inversion methods: pSp [33], e4e [37], restyle [5] and HFGI [41]. We first perform comparisons for the inversion of the StyleGAN2 model pre-trained on the FFHQ dataset. For each method we use the official implementation [32,38,4,42] to generate the results.

Qualitative Results Figure 4 shows the inversion results of the different methods. Overall, visual inspection shows that our method outperforms other

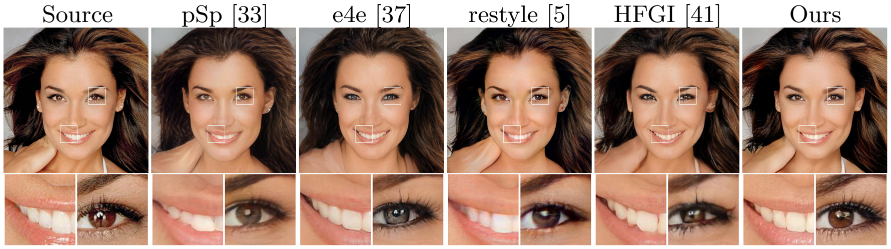


	Image	Patch	Image	Patch	Image	Patch	Image	Patch	Image	Patch
MSE ↓	0.012	0.013	0.019	0.014	0.012	0.009	0.009	0.011	**0.008**	**0.009**
LPIPS ↓	0.152	0.350	0.203	0.301	0.117	0.323	0.111	0.324	**0.066**	**0.201**

Fig. 4. Inversion of StyleGAN2 pretrained on face domain. We compare our model against state-of-the-art encoder-based methods [33,37,5,41] for the inversion of StyleGAN2 pre-trained on face domain. Our inversion results are visually more faithful and zoom-in patches show that they exhibit much more details and sharpness. Pixel-wise reconstruction errors (MSE error, lower is better) and perceptual quality (LPIPS distance, lower is better) confirm this visual conclusion on these examples

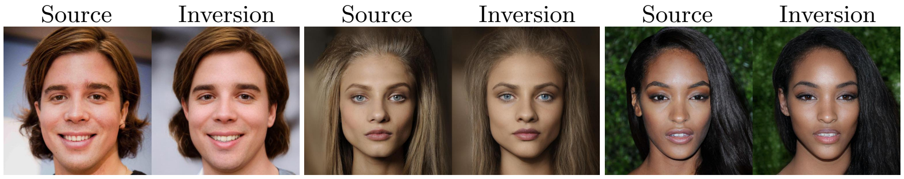


Fig. 5. Inversion of StyleGAN3 pretrained on face domain. We show preliminary inversion results of the 3rd generation of StyleGAN [21] on face domain. Compared with StyleGAN2, the architecture of StyleGAN3 has several important changes. Despite the architectural changes, our method still yields satisfying inversion results

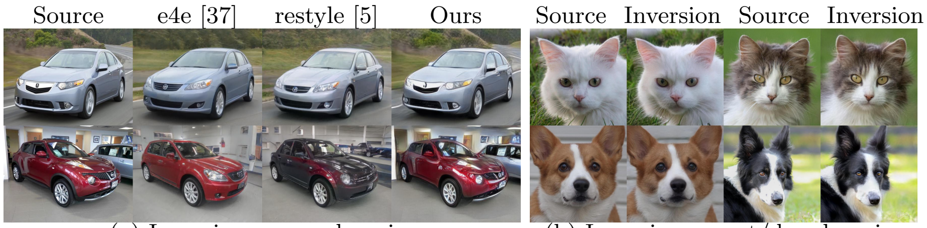
models. Firstly, faces are more faithfully reconstructed globally. Secondly, zoom-in patches show that more details are preserved and that the images produced by our framework are significantly sharper than those of the concurrent methods.

Inversion of StyleGAN3 We show preliminary inversion results of the third generation of StyleGAN [21] pretrained on FFHQ. Compared with StyleGAN2, the architecture of StyleGAN3 has several important changes. The input tensor passed into the generator is no longer constant, but synthesized from the latent code. The spatial size of the input tensor is increased from 4×4 to 36×36. As shown in Figure 5, despite the architectural changes, our proposed encoder still yields satisfying inversion results.

Quantitative Comparison We evaluate our approach quantitatively against the aforementioned encoder based methods [33,37,5,41] and a hybrid method (in-domain GAN) [50]. We compare each method on the inversion of StyleGAN2 pretrained on FFHQ, using the first $1K$ images of CelebA-HQ as evaluation data. To measure the reconstruction error, we compute *SSIM*, *PSNR* and *MSE*. To measure the perceptual quality, we measure the *LPIPS*[49] distance. Ad-

Table 1. Quantitative evaluation. We use *SSIM*, *PSNR* and *MSE* to measure the reconstruction error, and *LPIPS*[49] for the perceptual quality. We also measure the *identity similarity* (ID) between the inversion and the source image, which refers to the cosine similarity between the features in CurricularFace [15] of both images. To measure the discrepancy between the real data distribution and the inversion one, we use *FID* [13]. Overall, our method outperforms the state-of-the-art baselines by up to $10\% - 20\%$. In terms of perceptual quality (LPIPS), we improve the result by 50%

Method	SSIM ↑	PSNR ↑	MSE ↓	LPIPS ↓	ID ↑	FID ↓
IDGI[50]	0.554	22.06	0.034	0.136	0.18	36.83
pSp[33]	0.509	20.37	0.040	0.159	0.56	34.68
e4e[37]	0.479	19.17	0.052	0.196	0.51	36.72
restyle[5]	0.537	21.14	0.034	0.130	0.66	32.56
HFGI[41]	0.595	22.07	0.027	0.117	0.68	26.53
Ours	**0.641**	**23.65**	**0.019**	**0.066**	**0.80**	**19.03**



(a) Inversion on car domain.

(b) Inversion on cat/dog domain.

Fig. 6. Inversion on other domains. In (a), we show the inversion results of StyleGAN2 pre-trained on car domain. Our method captures better the details than e4e [37] and restyle-e4e [5]. In (b), we show the inversion results of StyleGAN2-Ada pre-trained on the cat and dog domains, respectively

ditionally, we measure the *identity similarity* (ID) between the inversion and the source image, which refers to the cosine similarity between the features in CurricularFace [15] of the two images. To measure the discrepancy between the real data distribution and the inversion one, we use the Frechet Inception Distance [13] (*FID*). Table 1 presents the quantitative evaluation of all the methods. Our method significantly outperforms the state-of-the-art methods on all the metrics. In terms of perceptual quality (LPIPS), improvement can attain 50%.

Inversion for other domains Figure 6(a) shows the inversion for StyleGAN2 pretrained on the car domain. We train the encoder with Stanford Car dataset [25]. Compared with e4e [37] and restyle-e4e[5], our inversion achieves a better reconstruction quality, preserving better the details. Figure 6(b) shows the inversion for StyleGAN2-Ada pretrained on AFHQ Cat/Dog dataset [9]. Our encoder achieves nearly perfect inversions. Here we did not compare with [37,5], as the official pre-trained model is unavailable.

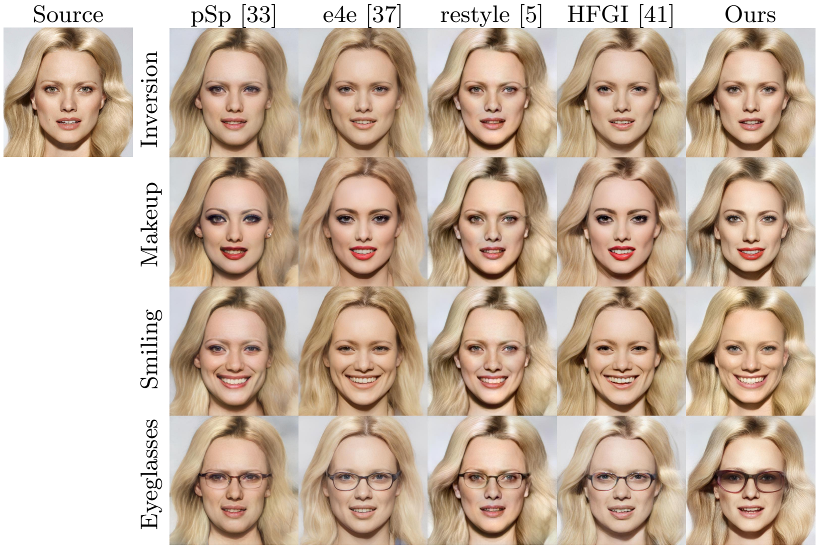


Fig. 7. Latent space editing. For each method, we apply InterFaceGAN [34] to perform latent editing for facial attribute manipulation. Our method yields plausible editing results, while at the same time preserving better the identity and the sharpness. More editing results are presented in the Supplementary Material

5.2 Editing

In this experiment, we consider the task of real image editing via latent space manipulation. We compare our approach with the state-of-the-art encoder-based GAN inversion methods [33,37,5,41] on the facial image editing via the latent space of StyleGAN2 pretrained on the FFHQ dataset. As such, for each inversion model, we generate the inverted latent codes for the first $10K$ images of CelebA-HQ, and use InterFaceGAN [34] to find the editing directions in the latent space. Figure 7 shows facial attribute editing results for all methods. Compared with the state-of-the-art, our method yields visually plausible editing results, while preserving better the identity and sharpness.

Additionally, we show style mixing results in Figure 8, generated from the latent code of one image with the feature tensor of another image. From this experiment we observe that the geometric structures such as pose and facial shape are encoded by the feature tensor, while the appearance styles like eye color and makeup are encoded by the latent code.

5.3 Video inversion

In this section, we discuss the possibility of integrating our proposed encoder into a video editing pipeline [47]. We compare the inversion quality and stability of different encoders on videos. Figure 9 shows the inversion results on several images extracted from the same video sequence. The last two frames are extreme

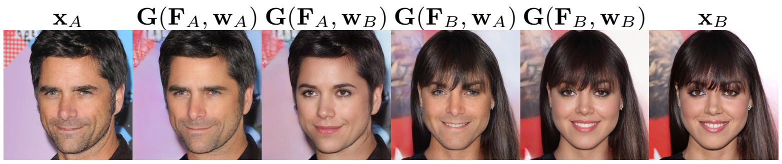

$\mathbf{x}_A$
$\mathbf{G}(\mathbf{F}_A, \mathbf{w}_A)$
$\mathbf{G}(\mathbf{F}_A, \mathbf{w}_B)$
$\mathbf{G}(\mathbf{F}_B, \mathbf{w}_A)$
$\mathbf{G}(\mathbf{F}_B, \mathbf{w}_B)$
$\mathbf{x}_B$

Fig. 8. Style mixing. The 2nd and 5th column show the inversions of two images $\mathbf{x}_A$ and $\mathbf{x}_B$, denoted by $\mathbf{G}(\mathbf{F}_A, \mathbf{w}_A)$ and $\mathbf{G}(\mathbf{F}_B, \mathbf{w}_B)$, respectively. The 3rd column is generated from the feature tensor of $\mathbf{x}_A$ and the latent code of $\mathbf{x}_B$, denoted by $\mathbf{G}(\mathbf{F}_A, \mathbf{w}_B)$, and vice versa for the 4th column, denoted by $\mathbf{G}(\mathbf{F}_B, \mathbf{w}_A)$. This shows that the feature tensor encodes the geometric structures such as pose and facial shape, whereas the latent code controls the appearance styles like eye color and makeup

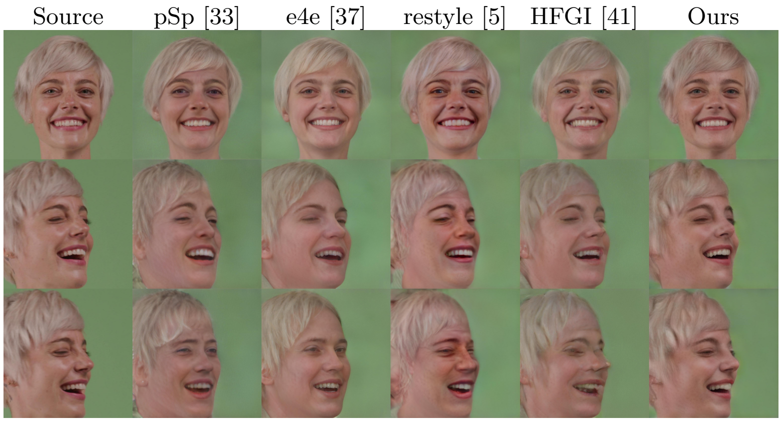


Fig. 9. Video inversion. For each method, we show the inversion results of several frames extracted from a video sequence. Our inversion method preserves better the identity along the video and yields a better reconstruction for the extreme poses

poses. As can be observed, other methods [33,37,5,41] fail to invert non-frontal poses, thus damaging the consistent reconstruction along the sequence. Our approach yields consistent inversion of high fidelity, which favors further editing on videos. Please refer to the supplementary material for the *video editing results*.

We evaluate our encoder quantitatively against the state-of-the-art for video inversion on RAVDESS [28], a dataset of talking face videos. From which we sample randomly 120 videos as evaluation data. For each method, we perform the inversion on each video and compute the quantitative metrics on the inversion results. As shown in Table 2, our approach outperforms the competing approaches on both the reconstruction error and the perceptual quality.

5.4 Ablation study

We conduct an ablation study on the training setup to analyze how each part of the losses contributes to the inversion quality. We compare the inversion quality

Table 2. Quantitative evaluation on video inversion. We sample randomly 120 videos from RAVDESS dataset[28], perform the inversion using each method and compute the quantitative metrics. Our method outperforms the competing approaches on both the reconstruction error and the perceptual quality

Method	SSIM ↑	PSNR ↑	MSE ↓	LPIPS ↓	ID ↑
pSp [33]	0.736	22.30	0.025	0.196	0.687
e4e [37]	0.713	20.57	0.037	0.220	0.620
restyle [5]	0.761	23.17	0.021	0.189	0.781
HFGI [41]	0.783	24.04	0.017	0.182	0.810
Ours	**0.818**	**26.64**	**0.009**	**0.122**	**0.895**

in the case of removing the per-pixel loss in eq.2, the identity loss in Eq. (6) and the face parsing loss in Eq. (7). We also compare the multi-scale perceptual loss in Eq. (3) to a normal LPIPS loss [49]. We further analyze the importance of the feature prediction branch and the choice of training data. Please refer to the supplementary material for the quantitative analysis of the ablation study.

5.5 Limitations

The main limitation of the proposed encoder lies in its global manipulation capacity. In the architecture of StyleGAN, the global attributes are controlled by lower layers, while the smaller local styles are controlled by higher layers. Our method yields better editing results on the attributes controlled by layers $> K$. To handle the attributes controlled by lower layers, we have proposed to modify the feature tensor using Eq. (1) to include the changes in the feature tensor. However, if the difference between the original feature tensor and the inverted one is important, this simple subtraction may generate artifacts. Moreover, the details reconstructed solely by the feature tensor are hard to change. In the future, it could be helpful to study further improvements for the feature tensor editing, *e.g.*, by including masks for the area of interest, or by training another network to generate the corresponding editing for the feature tensor.

6 Conclusion

In this paper, we propose a new encoder architecture for style-based GAN inversion and explore its editing capacity on images and videos. To the best of our knowledge, this is the first *feed-forward encoder* to include the feature tensor in the inversion, which significantly improves the perceptual quality of the inversion results, outperforming competitive state-of-the-art methods. Next, we introduce a *novel editing approach*, which makes the proposed encoder amenable to existing latent space editing methods. Experiments show that the editing capacity of our encoder is comparable to state-of-the-art methods while the editing results are of higher perceptual quality. Moreover, the experiments on video inversion show that our method yields a more accurate and stable inversion for videos. This could significantly facilitate *real-time editing in videos.*

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4432–4441 (2019)
2. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8296–8305 (2020)
3. Abdal, R., Zhu, P., Mitra, N., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. arXiv e-prints pp. arXiv–2008 (2020)
4. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Official implementation of restyle: A residual-based stylegan encoder via iterative refinement. `https://github.com/yuval-alaluf/restyle-encoder` (2021)
5. 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. pp. 6711–6720 (2021)
6. Alharbi, Y., Wonka, P.: Disentangled image generation through structured noise injection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5134–5142 (2020)
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=B1xsqj09Fm`
8. Chai, L., Zhu, J.Y., Shechtman, E., Isola, P., Zhang, R.: Ensembling with deep generative views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14997–15007 (2021)
9. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8188–8197 (2020)
10. Collins, E., Bala, R., Price, B., Susstrunk, S.: Editing in style: Uncovering the local semantics of gans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5771–5780 (2020)
11. 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)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
13. 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)
14. Hou, X., Zhang, X., Liang, H., Shen, L., Lai, Z., Wan, J.: Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing. Neural Networks (2021)
15. Huang, Y., Wang, Y., Tai, Y., Liu, X., Shen, P., Li, S., Li, J., Huang, F.: Curricularface: adaptive curriculum learning loss for deep face recognition. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5901–5910 (2020)
16. Huh, M., Zhang, R., Zhu, J.Y., Paris, S., Hertzmann, A.: Transforming and projecting images into class-conditional generative networks. In: European Conference on Computer Vision. pp. 17–34. Springer (2020)

17. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: Proc. NeurIPS (2020)
18. Kang, K., Kim, S., Cho, S.: Gan inversion for out-of-range images with geometric transformations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13941–13949 (2021)
19. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (2018)
20. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Proc. NeurIPS (2020)
21. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021)
22. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
23. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020)
24. Kim, H., Choi, Y., Kim, J., Yoo, S., Uh, Y.: Exploiting spatial dimensions of latent in gan for real-time image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 852–861 (2021)
25. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
26. Kwon, G., Ye, J.C.: Diagonal attention and style-based gan for content-style disentanglement in image generation and translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13980–13989 (2021)
27. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: Editgan: High-precision semantic image editing. arXiv preprint arXiv:2111.03186 (2021)
28. Livingstone, S.R., Russo, F.A.: The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. PloS one **13**(5), e0196391 (2018)
29. Park, T., Zhu, J.Y., Wang, O., Lu, J., Shechtman, E., Efros, A., Zhang, R.: Swapping autoencoder for deep image manipulation. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 7198–7211. Curran Associates, Inc. (2020), `https://proceedings.neurips.cc/paper/2020/file/50905d7b2216bfeccb5b41016357176b-Paper.pdf`
30. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2085–2094 (2021)
31. Pidhorskyi, S., Adjeroh, D.A., Doretto, G.: Adversarial latent autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14104–14113 (2020)
32. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Official implementation of encoding in style: a stylegan encoder for image-to-image translation. `https://github.com/eladrich/pixel2style2pixel` (2020)
33. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2287–2296 (2021)

34. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9243–9252 (2020)
35. 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)
36. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.P., Pérez, P., Zollhofer, M., Theobalt, C.: Stylerig: Rigging stylegan for 3d control over portrait images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6142–6151 (2020)
37. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG) **40**(4), 1–14 (2021)
38. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Official implementation of designing an encoder for stylegan image manipulation. `https://github.com/omertov/encoder4editing` (2021)
39. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the gan latent space. In: International Conference on Machine Learning. pp. 9786–9796. PMLR (2020)
40. Wang, B., Ponce, C.R.: The geometry of deep generative image models and its applications. arXiv preprint arXiv:2101.06006 (2021)
41. Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: High-fidelity gan inversion for image attribute editing. arXiv preprint arXiv:2109.06590 (2021)
42. Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: Official implementation of high-fidelity gan inversion for image attribute editing. `https://github.com/Tengfei-Wang/HFGI` (2021)
43. Wei, T., Chen, D., Zhou, W., Liao, J., Zhang, W., Yuan, L., Hua, G., Yu, N.: A simple baseline for stylegan inversion. arXiv preprint arXiv:2104.07661 (2021)
44. 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)
45. Xia, W., Zhang, Y., Yang, Y., Xue, J.H., Zhou, B., Yang, M.H.: Gan inversion: A survey. arXiv preprint arXiv:2101.05278 (2021)
46. Xu, Y., Du, Y., Xiao, W., Xu, X., He, S.: From continuity to editability: Inverting gans with consecutive images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13910–13918 (2021)
47. Yao, X., Newson, A., Gousseau, Y., Hellier, P.: A latent transformer for disentangled face editing in images and videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13789–13798 (2021)
48. Yu, C., Wang, W.: Adaptable gan encoders for image reconstruction via multi-type latent vectors with two-scale attentions. arXiv preprint arXiv:2108.10201 (2021)
49. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
50. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: European Conference on Computer Vision. pp. 592–608. Springer (2020)
51. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Barbershop: Gan-based image compositing using segmentation masks. arXiv preprint arXiv:2106.01505 (2021)
52. zllrunning: Face parsing network pre-trained on celebamask-hq dataset. `https://github.com/zllrunning/face-parsing.PyTorch` (2019)