Gradual Residuals Alignment: A Dual-Stream Framework for GAN Inversion and Image Attribute Editing

Hao Li[1], Mengqi Huang[1], Lei Zhang[1], Bo Hu[1], Yi Liu[2], Zhendong Mao[1]*

[1]University of Science and Technology of China, Hefei, China
[2]State Key Laboratory of Communication Content Cognition, Beijing, China
{lihaohn, huangmq}@mail.ustc.edu.cn, {leizh23, hubo, zdmao}@ustc.edu.cn, gavin1332@gmail.com

Abstract

GAN-based image attribute editing firstly leverages GAN Inversion to project real images into the latent space of GAN and then manipulates corresponding latent codes. Recent inversion methods mainly utilize additional high-bit features to improve image details preservation, as low-bit codes cannot faithfully reconstruct source images, leading to the loss of details. However, during editing, existing works fail to accurately complement the lost details and suffer from poor editability. The main reason is they inject all the lost details indiscriminately at one time, which inherently induces the position and quantity of details to overfit source images, resulting in inconsistent content and artifacts in edited images. This work argues that details should be gradually injected into both the reconstruction and editing process in a multi-stage coarse-to-fine manner for better detail preservation and high editability. Therefore, a novel dual-stream framework is proposed to accurately complement details at each stage. The Reconstruction Stream is employed to embed coarse-to-fine lost details into residual features and then adaptively add them to the GAN generator. In the Editing Stream, residual features are accurately aligned by our Selective Attention mechanism and then injected into the editing process in a multi-stage manner. Extensive experiments have shown the superiority of our framework in both reconstruction accuracy and editing quality compared with existing methods.

1 Introduction

Image attribute editing, which aims to modify the desired attributes of a given image while preserving other details, has gained increasing research interest for its various real-world applications. Rapid progress has been made in this area with the development of generative adversarial network (GAN) based editing methods, which leverage the latent space of pre-trained GAN models (typically, $\mathcal{W}+$ of StyleGAN) by GAN Inversion (Abdal, Qin, and Wonka 2019). The critical challenge of GAN Inversion lies in achieving the unity of both high-fidelity details preservation and high editing quality since the distortion-editability trade-off (Tov et al. 2021).

Early GAN Inversion methods (Richardson et al. 2021; Tov et al. 2021) focus on *Low-bit Inversion* to better map images to low-bit codes (*i.e.*, low-dimension latent codes $\in \mathbb{R}^{18\times512}$), resulting in severe details lost in both of their reconstructed and edited images compared to the source images. Recent works introduced *High-bit Inversion*, which first utilize low-bite codes to generate coarse results and further use high-bit features (*e.g.*, high-dimension feature maps $\in \mathbb{R}^{64\times64\times512}$) to improve details preservation. These high-bit features are derived from source images or the residuals between source images and low-bit codes' reconstruction. For example, (Yao et al. 2022; Liu, Song, and Chen 2023) propose to replace a part of low-bit codes with high-bit features and then restart generation. (Wang et al. 2022; Pehlivan, Dalva, and Dundar 2023) focus on establishing an additional branch to complement details for image generation by calculating residuals. In conclusion, most recent works pur-

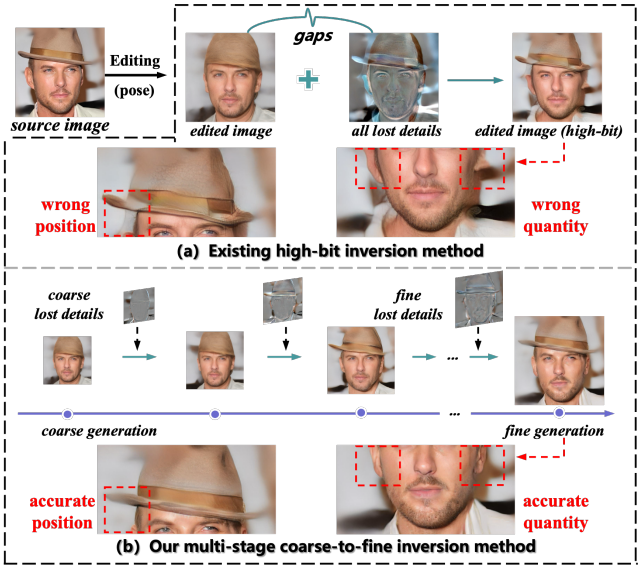

Figure 1: Illustration of our motivation. Giving a source image and then editing (e.g., pose) it. (a) Existing high-bit inversion injects lost details of reconstruction into the edited images as much as possible at one time, which leads to inconsistent content and artifacts. (b) Our method gradually aligns and complements lost details at different stages in editing, which achieves a unity of both high-quality details preservation and high editability with the artifacts mitigated.

*Corresponding author.

sue injecting source images' high-bit features into the edited images *as much as possible at one time*.

However, existing works overlook the intricate gaps between source images and edited images, and therefore fail to *accurately* complement the lost details for the edited images, leading to poor editability and incoherent generation results. The editing operation itself will bring more or less variations to source images from global layouts to local patterns, and therefore the lost details desired by edited images are also changed synchronously. As shown in Fig.1 (a), the variation of pose leads to differences in details' target position (e.g., the hat) and details' target quantity (e.g., the left ear should have fewer details while the right ear should have more details than the source image). Existing methods inject all the lost details indiscriminately, inherently inducing the position and quantity of details to overfit source images, resulting in inconsistent content and artifacts.

This paper argues that in both reconstruction and editing, lost details should be gradually complemented in a multi-stage coarse-to-fine manner for accurate detail preservation and compelling editability. The reason is that both the reconstruction and editing themselves are, by nature, coarse-to-fine processes with the scale of feature maps increasing, and different-granularity details are required step by step. Gradual addition offers two distinct advantages: (1) Complementing coarse-to-fine details at each step results in cumulative benefits and improves the overall reconstruction quality. (2) The position and quantity of coarse details are easier to align with edited images, which provides a better association foundation for the alignment of finer details, thereby reducing the overall difficulty of the editing. Take Fig.1 (b) as an example. With the coarse-to-fine lost details gradually injected, the content of the edited image becomes closer to the source (*e.g.*, the texture of the hat and ears gradually match the source image), meanwhile, the position and quantity of details will be more accurate (*e.g.*, position of the hat, quantity of ears). This manner can effectively mitigate the issue of artifacts and achieve the unity of both high-fidelity detail preservation and high editing quality.

With this motivation, we propose a novel framework named **G**radual **R**esiduals **A**lignment **D**ual-Stream Framework for **Style**GAN inversion and editing (**GradStyle**), which effectively extracts coarse-to-fine lost details for faithful reconstruction and accurately aligns them with edited images for flexible editing in a multi-stage manner. Specifically, this framework includes a Reconstruction Stream and an Editing Stream, with an Encoding Phase for embedding images. GAN generator blocks are grouped into coarse-to-fine consecutive stages based on their characteristics. In Reconstruction Stream, proposed *Gradual Residual Module* embeds the feature-level distortions between the coarsely reconstructed images and source images into multiple residual features to complement lost details at each stage. A gated fusion mechanism with regularization is further utilized to adaptively fuse residual features in a learnable manner. In the Editing Stream, we propose a novel *Global Alignment Module*, which first achieves an accurate global alignment for residual features based on our *Selective Attention* mechanism, and then adaptively injects them into the editing process. This global alignment provides an effective adjustment for the position and quantity of lost details. To simultaneously train both streams, a self-supervised training strategy without additional labeled edited images is devised.

Our main contributions are summarized as follows:

- For the first time, we propose a scheme to gradually complement lost details in the reconstruction and editing of images, which achieves a unity of both high-quality detail preservation and high editability in StyleGAN.
- A novel dual-stream framework, *i.e.,* GradStyle, is proposed to simultaneously conduct reconstruction and editing with a devised self-supervised training strategy. Reconstruction Stream explores coarse-to-fine details information and achieves a more faithful reconstruction, while Editing Stream accurately aligns and adaptively injects these details into the editing process step by step, ensuring better editability.
- Extensive experiments have shown the effectiveness of our framework and the improvement over existing methods in terms of both reconstruction accuracy and editing quality, with the generalizability toward various domains.

2 Related Works

2.1 GAN Inversion

GAN Inversion is to invert a given image back into the latent space of a pre-trained GAN model and obtain a latent representation with the capacity to reconstruct it. Recently, the StyleGAN series (Karras, Laine, and Aila 2019; Karras et al. 2020b,a, 2021) have gained widespread popularity due to its fantastic disentangled latent space which facilitates attribute editing. Our study mainly focuses on StyleGAN inversion.

Low-bit Inversion. Early optimization-based approaches (Zhu et al. 2016; Huh et al. 2020; Abdal, Qin, and Wonka 2020) continuously optimize the latent codes to minimize the reconstruction loss of the source with slow inference. Encoder-based approaches (Richardson et al. 2021; Tov et al. 2021; Hu et al. 2022; Mao et al. 2022) map latent codes more quickly through a learnable encoder, with better editability but worse fidelity. To keep more details, (Wei et al. 2022; Moon and Park 2022) complement latent codes with the differences between reconstructed and source images. Other works (Roich et al. 2022; Dinh et al. 2022) have attempted to fine-tune the generator. These methods all fail to maintain details due to using low-bit codes only.

High-bit Inversion. BDInvert (Kang, Kim, and Cho 2021) first proposes using an additional latent space $\mathcal{F}$. HFGI (Wang et al. 2022) utilizes the image-level distortions between the source and the reconstructed, but image-level features retain excessive spatial dependencies. Other methods (Yao et al. 2022; Liu, Song, and Chen 2023) train an encoder to obtain low-bit codes and high-bit features simultaneously, then replace the first several latent codes with high-bit features. StyleRes (Pehlivan, Dalva, and Dundar 2023) actually packs all details information into a single residual feature and adds it at one stage. The above methods suffer from severe artifacts, however, our approach, which employs a multi-stage manner to gradually complement coarse-to-fine details, can effectively suppress artifacts.

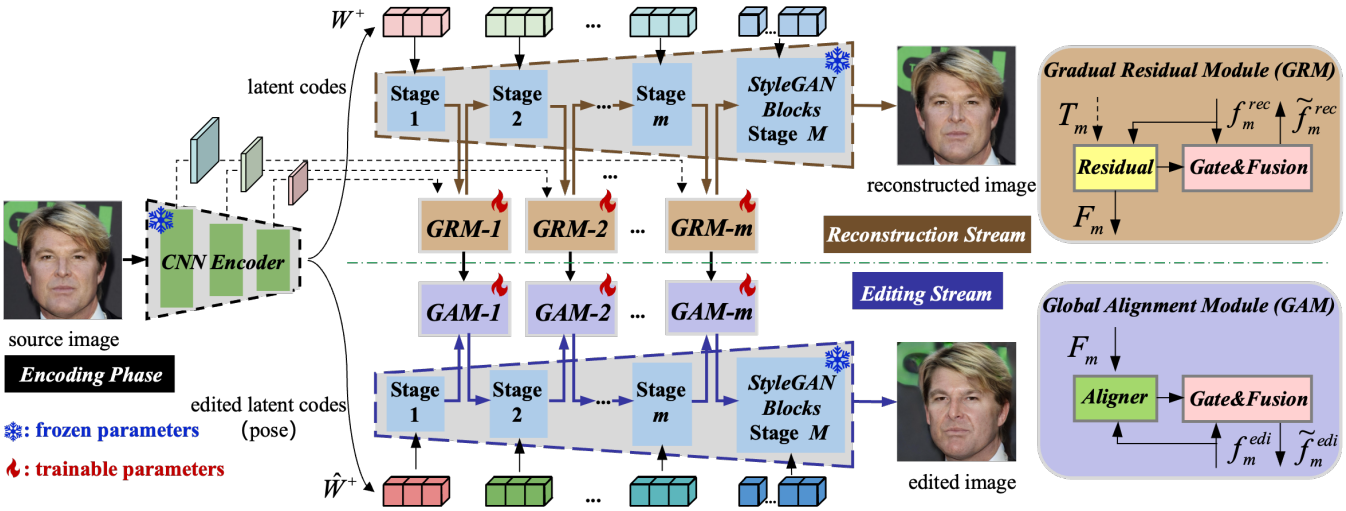

Figure 2: An overview of our dual-stream framework GradStyle. It consists of three parts, an Encoding Phase for embedding images, a Reconstruction Stream for faithful reconstruction and residual features calculation, and an Editing Stream for edited image generation by gradually aligning and adding details information. The proposed Gradual Residual Module and Global Alignment Module are also illustrated, and details of Aligner are especially shown in Fig.3.

2.2 Latent Space Editing

Manipulating latent codes in the latent space focuses on searching meaningful editing directions for interpolation. For supervised methods, off-the-shelf attribute classifiers are employed to obtain attribute labels and then analyze the spatial distribution of latent codes for editing directions. For example, InterfaceGAN (Shen et al. 2020b) utilizes Support Vector Machines (SVMs) to learn a classification hyperplane. (Abdal et al. 2021; Wang, Yu, and Fritz 2021) employs neural networks to distinguish directions between different attributes. For unsupervised methods, GANspace (Härkönen et al. 2020) applies Principal Component Analysis (PCA), (Shen and Zhou 2021) decomposes the model weights of GAN networks. Moreover, with the advancement of multimodal techniques (Radford et al. 2021), language-based methods (Wu, Lischinski, and Shechtman 2021; Patashnik et al. 2021) have further expanded the application area.

3 Methodology

The overall framework is depicted in Fig.2. In the Encoding Phase (section 3.1), an encoder is adopted to embed source images to both low-bit latent codes and hierarchical features. Reconstruction Stream (section 3.2) employs *Gradual Residual Module (GRM)* to calculate residual features and faithfully reconstruct images. In Editing Stream (section 3.3), we utilize *Global Alignment Module (GAM)* to align residual features with edited images and inject them into the generator step by step. Finally, a self-supervised training strategy (section 3.4) is conducted to train two streams simultaneously. Next, we will describe them in detail.

Notations. Formally, $X \in \mathbb{R}^{H_0 \times W_0 \times 3}$ denotes source images. Both reconstruction and editing utilize the same generator from pre-trained StyleGAN but are driven by different latent codes. N is the number of latent codes of each image, M is the total number of generation stages, T_m and F_m respectively denote the hierarchical features from the encoder and the residual features from *GRM* at stage m. X^r and X^e is reconstructed and edited images, f_m^{rec} and f_m^{edi} denote the corresponding feature maps of generator blocks at stage m.

3.1 Encoding Phase

With the pre-trained CNN encoder E_0 from (Tov et al. 2021), we have $T, W^+ = E_0(X)$, where $W^+ = \{w_i | i = 1, 2, \cdots, N,\ w_i \in \mathbb{R}^{512}\}$ are latent codes and can coarsely reconstruct X. Our encoder is based on a pyramid structure to generate latent codes, corresponding hierarchical features $T = \{T_m | m = 1, 2, \cdots, M\}$ can be naturally obtained from the different layers of the hierarchical encoder, where $T_m \in \mathbb{R}^{H_m \times W_m \times c}$, $(H_m, W_m) = (H_0/n_m, W_0/n_m), c = 512$. These hierarchical features represent the coarse-to-fine details information of the source images. Further, for our editing stream, as W^+ can be interpolated by meaningful direction (Shen et al. 2020a), we obtain edited latent codes $\hat{W}^+ = W^+ + \alpha \Delta W^+$, where ΔW^+ is the editing direction and α is the editing amplitude.

3.2 Reconstruction Stream

This stream targets a faithful reconstruction with the input of W^+ and T, and calculating residual features F_m for the Editing Stream. Each w_i controls a StyleGAN block, i.e., Modulated Convolution layer (Karras et al. 2020b), and the different block affects different content from coarse (e.g., shapes of face) to fine levels (e.g., wrinkles). Instead of naively refining all blocks, we propose to selectively refine several key blocks (which are enough to complement all lost details) to further improve efficiency. Specifically, as shown in Fig.2, all generator blocks with corresponding latent codes are grouped into coarse-to-fine consecutive M

parts, and each part is treated as a generation stage. We then insert a *GRM* between every two stages. At stage m, with hierarchical features T_m and the output of generator block f_m^{rec}, we calculate that:

$$\tilde{f}_m^{rec}, F_m = GRM_m(T_m, f_m^{rec}), \tag{1}$$

where F_m, f_m^{rec} and $\tilde{f}_m^{rec} \in \mathbb{R}^{H_m \times W_m \times c}$, and $\tilde{f}_m^{rec}$ includes richer details than f_m^{rec}, thereby serving as the input feature of the next stage. After crossing all stages, we can obtain a well-reconstructed image.

Gradual Residual Module (GRM) and Gate&Fusion. It is essential to find out what details the current stage fails to reconstruct concerning the source image, and then complement them, so we design *Gradual Residual Module*. In each *GRM*, we utilize the ResNet-based network E_{res} to obtain the residual features between T_m and f_m^{rec},

$$F_m = E_{res}(W_T T_m, W_f f_m^{rec}), \tag{2}$$

for simplicity, we employ W_T and W_f to denote learnable convolution networks, which transfer both features to the same semantic space. Further, we utilize the *Gate&Fusion* to learn how to adaptively fuse the residual features, as not all details are required to be added at this stage. We need to choose them in a gating manner according to the characteristics of different stages:

$$g_m = \sigma(W_g\,[f_m^{rec}, F_m]), \tag{3}$$

$$\tilde{f}_m^{rec} = f_m^{rec} + g_m \cdot F_m, \tag{4}$$

where $\sigma(\cdot)$ is a *sigmoid* function, W_g is the learnable layers. The gating maps g_m share the same size with F_m and can determine which patches of F_m are used at this stage. Residual features can be adaptively selected by *Gate&Fusion*, so an extra L_1 *Regularization* term is utilized to avoid the overfitting of details stemming from redundant information:

$$\mathcal{L}_f = \sum_{m=1}^{M} \left\| \tilde{f}_m^{rec} - f_m^{rec} \right\|_1 = \sum_{m=1}^{M} \|g_m \cdot F_m\|_1 . \tag{5}$$

3.3 Editing Stream

Generator blocks of this stream are grouped in the same way as the Reconstruction Stream, and each *GAM* is also inserted between two stages. With the edited latent codes $\hat{W}^+$, each stage outputs an edited feature map f_m^{edi}. Our *GAM* receives the residual features F_m from *GRM* at each stage and then aligns F_m with f_m^{edi} through *Aligner* block, finally getting $\tilde{f}_m^{edi}$ which will include aligned lost details. That is

$$\tilde{f}_m^{edi} = GAM_m(F_m, f_m^{edi}). \tag{6}$$

After the aligned details information is added across the whole generation process, the edited image with more accurate and consistent details can be obtained.

Global Alignment Module (GAM) and Selective Attention. *GAM* is aimed to achieve a more accurate global alignment according to the semantic correlation between unaligned residual features F_m and edited feature map f_m^{edi}. The main basis is that F_m inherently inherits the semantic characteristics of reconstructed image feature map f_m^{rec},

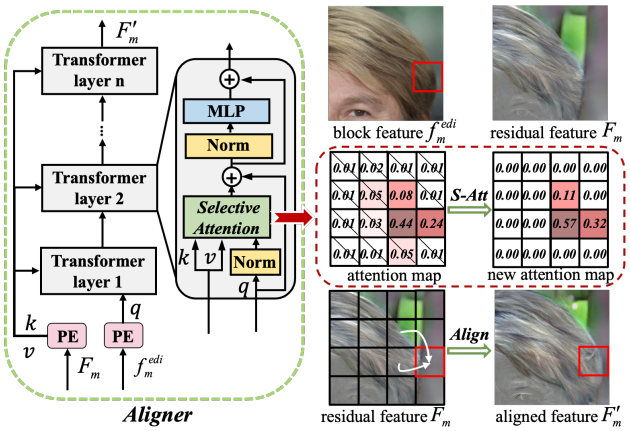

Figure 3: Detailed structure of Aligner block and an image-level visualization example for Selective Attention (we actually utilize it in the feature level). In the 1st row of the example, a coarsely edited image stands for the block feature f_m^{edi} (query), and the unaligned residual feature F_m (key and value) is on its right. In the 2nd row, for a region of query, its attention map indicates that there are many irrelevant regions, Selective Attention will suppress irrelevant regions and enhance relevant regions. The last row shows that a region of $F_m^{'}$ is combined by similar regions of unaligned F_m.

which has a strong content-level correlation with f_m^{edi}. So we design a novel attention-based *Aligner* block for this alignment, as shown in Fig.3. Thanks to the Transformer structure, we can deal with various editing scenes no matter how the position and quantity of details change, as this structure owns the long-range awareness to associate and fuse similar region features. *Aligner* is represented as

$$F_m^{'} = E_{ab}(F_m, f_m^{edi}), \tag{7}$$

where $F_m^{'}$ is the aligned residual features and E_{ab} is the *Aligner* block at each stage.

In details, the input feature F_m and $f_m^{edi} \in \mathbb{R}^{H_m \times W_m \times c}$ are first flattened to $\mathbb{R}^{L \times c}$, where $L = H_m \times W_m$, and then both go through a Sinusoidal Positional Embedding layer *PE* (Vaswani et al. 2017), getting Z_F, Z_f. The key component for alignment is our *Selective Attention* mechanism and we apply it several times for fully exploring semantic correlation. It can be mathematically formed as:

$$q, k, v = W_q Z_f, W_k Z_F, W_v Z_F, \tag{8}$$

$$S\text{-}Att(Z_F, Z_f) = \mathrm{Softmax}\left(\frac{qk^T \odot \mathrm{Top}_\mu(qk^T)}{\sqrt{d_k}}\right)v, \tag{9}$$

where W_q, W_k, W_v are the learnable parameters, scaling factor $d_k = 64$, and the multi-head mechanism is employed. Eq.9 indicates that only the top $\mu\%$ values of qk^T will undergo the $\mathrm{Softmax}$ operation to calculate the attention map, while the remaining values will be suppressed, as shown in Fig.3. Based on our *Selective Attention*, *Aligner* can not only align F_m by combining the relevant regions but also weaken the influence of irrelevant regions. Following *GRM*, our *GAM* also employ *Gate&Fusion* block to adaptively fuse features, and finally get $\tilde{f}_m^{edi}$.

3.4 Training

Self-supervised Training. During training, the encoder and StyleGAN2 generator blocks are all fixed, the key is how to seamlessly train our *GRMs* and *GAMs*. All *GRMs* will be updated under the guidance of the reconstruction error of the source X, with the gradient from *GAMs* detached. For the Editing Stream, alleviating the misalignment between residual features and edited feature maps is the ultimate goal of *GAMs*. The training of *GAMs* necessitates enough misaligned feature pairs, but the absence of manually annotated edited images precludes the availability of ground truth.

A self-supervised training strategy is devised for our Editing Stream. We set $\alpha = 0$ in $\hat{W}^+ = W^+ + \alpha\Delta W^+$ in the Encoding Pharse, which implies Editing Stream will generate the same output images as the Reconstruction Stream and can also calculate loss based on the source X. As the same latent codes result in the same generator block features, received residual features F_m will become well-aligned with the edited image feature maps f_m^{edi}. To train their alignment ability, we augment F_m with random perspective transformation (Wang et al. 2022) to simulate the layout misalignment with f_m^{edi}, that is $\hat{F}_m = \mathrm{Trans}(F_m)$, thereby getting misaligned feature pairs $\{\hat{F}_m, f_m^{edi}\}$ with the ground truth F_m. The *Aligner* block is encouraged to produce aligned residual features $F_m^{'} = E_{ab}(\hat{F}_m, f_m^{edi})$. We take F_m as the intermediate supervision signal and utilize an aligner loss:

$$\mathcal{L}_{aligner} = \sum_{m=1}^{M} \left\| F_m^{'} - F_m \right\|_1 . \tag{10}$$

For the consistency of feature discrimination abilities, *Gate&Fusion* block in *GAMs* share weights with *GRMs'*.

Training Losses. The source is X, and the output of two streams are X^r and X^e. Following (Tov et al. 2021), we first employ L_2 loss, $lpips$ loss for faithful reconstruction:

$$\mathcal{L}_{l2} = \|X^r - X\|_2 + \|X^e - X\|_2 , \tag{11}$$

$$\mathcal{L}_{lpips} = \|\Phi(X^r) - \Phi(X)\|_2 + \|\Phi(X^e) - \Phi(X)\|_2 , \tag{12}$$

where $\Phi(\cdot)$ is the pre-trained VGG network (Simonyan and Zisserman 2014). ID loss is to keep the identity consistent,

$$\mathcal{L}_{id} = (1 - \langle F(X), F(X^r)\rangle) + (1 - \langle F(X), F(X^e)\rangle), \tag{13}$$

where $F(\cdot)$ is pre-trained ArcFace (Deng et al. 2019) or a ResNet for different domains (Tov et al. 2021).

For better image quality, we also utilize an adversarial loss $\mathcal{L}_{adv}$ based on a discriminator D, Which is initialized with well-trained parameters from StyleGAN2 and then trains along with our framework.

$$\mathcal{L}_{adv} = -\mathbb{E}[log(D(X^r))] - \mathbb{E}[log(D(X^e))]. \tag{14}$$

The overall loss is a weighted sum of the above losses:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{l2} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{id}\mathcal{L}_{id} + \lambda_{adv}\mathcal{L}_{adv} \\ + \lambda_f \mathcal{L}_f + \lambda_{aligner}\mathcal{L}_{aligner}. \end{aligned} \tag{15}$$

See the Appendix for hyperparameters and more details. After training, our framework can inverse images through the Reconstruction Stream and meanwhile conduct attribute editing in the Editing Stream.

4 Experiments

4.1 Settings

Experimental setup. Our approach is based on pre-trained StyleGAN2 (Karras et al. 2020b) and e4e encoder (Tov et al. 2021). Main experiments are conducted in the face-domain dataset, we use the FFHQ (Karras, Laine, and Aila 2019) to train and the Celeba-HQ (Karras et al. 2017) to evaluate. During image editing, we choose off-the-shelf InterfaceGAN (Shen et al. 2020b) and GANspace (Härkönen et al. 2020) as latent code editors. For generalizability evaluation, we also test our method in the different domain datasets, including Stanford Car (Krause et al. 2013) for car and Metface (Karras et al. 2020a) for artistic portrait.

Implementation details. Our framework is mainly trained on face-domain images with 1024×1024 resolution ($N = 18$ latent codes in total), adopting Adam optimizer (Kingma and Ba 2014) with LookAhead technique (Zhang et al. 2019). In all experiments, both streams of our framework are 4-stage processes, which means the 3rd, 6th, and 8th blocks of the StyleGAN generator are refined. Other details are included in our Appendix.

4.2 Evaluations

Quantitative results. To verify the reconstruction fidelity and editability, we compare our method quantitatively with other state-of-the-art methods. Low-bit Inversion includes e4e (Tov et al. 2021), ReStyle (Alaluf, Patashnik, and Cohen-Or 2021) and HyperStyle (Alaluf et al. 2022), for High-bit Inversion, HFGI (Wang et al. 2022), StyleRes (Pehlivan, Dalva, and Dundar 2023) and CLCAE (Liu, Song, and Chen 2023) have been included. We compare them by reporting some metrics that are calculated on the highest resolution of the first 1000 images from Celeba-HQ. We adopt L2 distance, LPIPS (Zhang et al. 2018), and SSIM (Wang et al. 2004) to measure pixel-level, feature-level, and structure-level similarity between source and reconstructed images. A pre-trained identity-recognition network (Huang et al. 2020) is employed to measure identity similarity (ID), and we also report the Peak Signal-to-Noise Ratio (PSNR).

As shown in our Table.1, we report quantitative comparisons of reconstruction quality. High-bit Inversion has better results than Low-bit Inversion on almost all metrics. Our method achieves the best results among all competing methods on all metrics, implying our coarse-to-fine strategy helps to generate richer and more accurate image content. Most significantly, we have achieved a significant improvement in identity preservation, which means our method has a better ability to keep identity details.

In Table.2, we show quantitative comparisons of attribute editing. As the smile always involves the quantity change of details while the pose involves the position change, we choose them as representatives. We add or remove the smile and pose in our test images, and then calculate the average ID score as the straight quantitative measurement to evaluate editing performance since other metrics are no longer suitable for editing. It shows our method works better in attribute control, implying that we have flexibly manipulated the special attribute with enough details preserved.

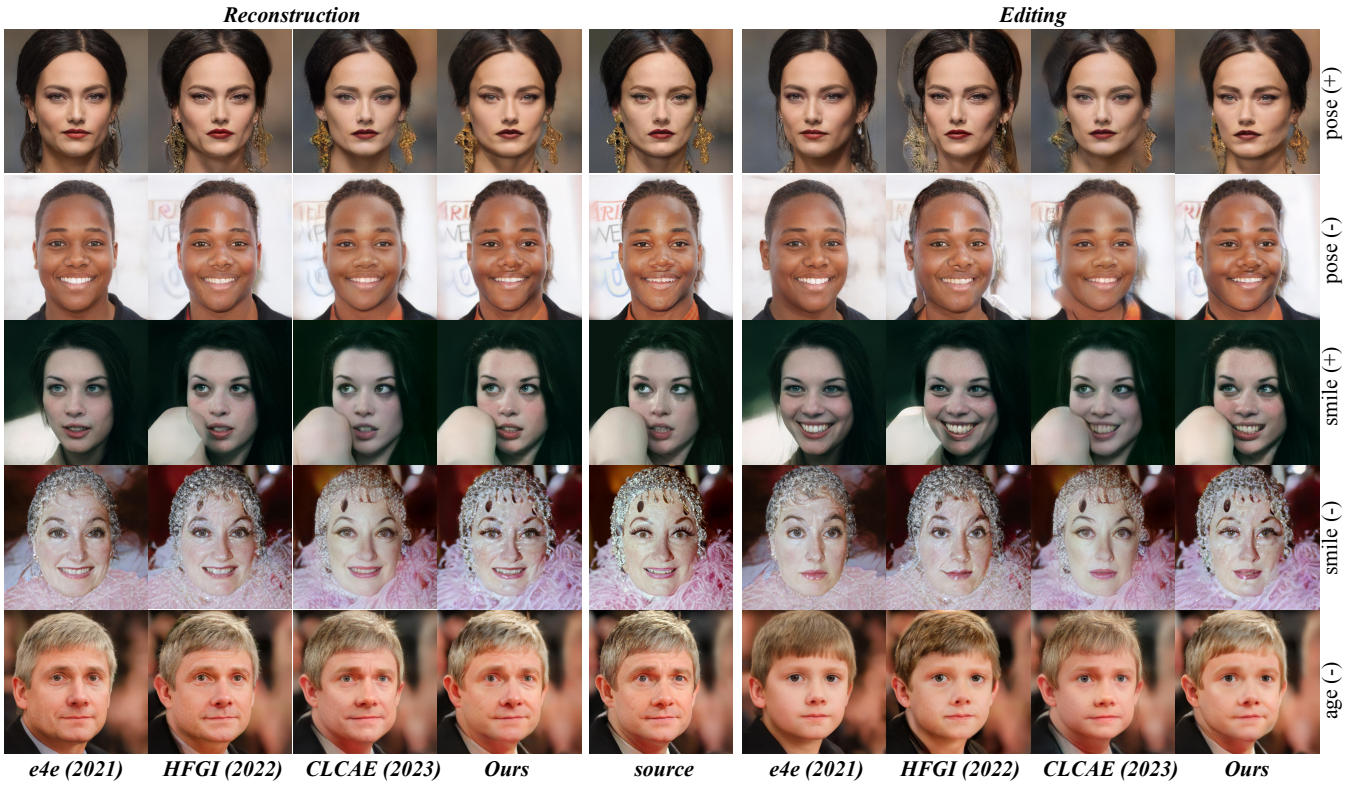

Figure 4: Qualitative results of reconstruction and editing. The left shows reconstructed results from several recent methods and our method, and the right shows edited results based on InterfaceGAN (Shen et al. 2020b). The source is in the middle.

Method	ID(↑)	SSIM(↑)	L2(↓)	LPIPS(↓)	PSNR(↑)
e4e	0.499	0.605	0.053	0.394	19.124
ReStyle	0.506	0.607	0.049	0.384	19.462
HyperStyle	0.697	0.627	0.035	0.352	21.023
HFGI	0.682	0.645	0.027	0.328	22.065
StyleRes	0.758	0.674	0.019	0.286	23.603
CLCAE	0.719	0.687	0.016	0.289	24.375
GradStyle (ours)	**0.813**	**0.696**	**0.015**	**0.269**	**24.583**

Table 1: Quantitative comparisons of reconstruction quality, ↓ indicates lower is better while ↑ indicates higher is better.

Method	Pose(+)	Pose(-)	Smile(+)	Smile(-)
	ID(↑)			
e4e	0.464	0.461	0.446	0.379
ReStyle	0.487	0.487	0.468	0.428
HyperStyle	0.641	0.651	0.608	0.577
HFGI	0.556	0.541	0.544	0.480
StyleRes	0.581	0.584	0.583	0.556
CLCAE	0.675	0.672	0.653	0.637
GradStyle (ours)	**0.677**	**0.689**	**0.690**	**0.671**

Table 2: Quantitative comparisons of attribute editing. (+) stands for adding this attribute while (-) stands for removing.

Qualitative results. To visually demonstrate the advantages of our method, we have compared it with other three recent representative methods in Fig.4: e4e (Tov et al. 2021), HFGI (Wang et al. 2022) and CLCAE (Liu, Song, and Chen 2023). We add or remove three types of facial attributes: pose, smile, and age in human faces, and show their reconstructed images on the left of Fig.4 while the edited images are on the right.

We modify the pose of faces in the first two rows of Fig.4. We can see that e4e easily edits images but loses many details, such as earrings (1st row) and the background (2nd row). HFGI suffers from reconstruction errors and severe silhouette *artifacts* in editing. CLCAE can keep the most details in reconstruction but it fails to flexibly edit (1st and 2nd row). Better than all, our method can correctly align lost details without *artifacts*. In 3rd row, our method generates more natural teeth with the arm preserved but other methods fail to do both. For other rows, we can observe that our method preserves more details in both reconstruction and editing, such as more faithful clothes and headwear (4th row), and the more similar hairstyle and face (5th row). In short, our method achieves the most faithful reconstruction and the best editing quality among these methods.

Generalizability. In this work, we use a self-supervised training strategy to train our framework without any labeled edited images. However, what inspires us is that the performance of our framework in the manipulations of various attributes (such as nose, lipstick, lighting, mascara, and eyes) is also plausible, as shown in Fig.5. These editings are based

Figure 5: Generalizability of our self-supervised training strategy to deal with various attributes.

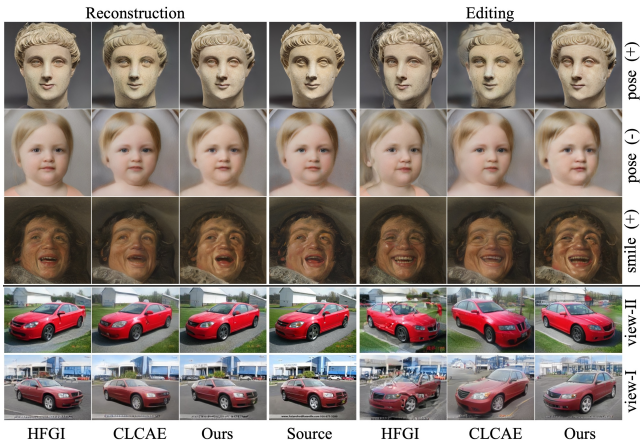

Figure 6: Generalizability of our method in different domains with reconstruction (left) and editing (right).

on InterfaceGAN (Shen et al. 2020b). It implies that the self-supervised training strategy can train our framework's universal editing capability regardless of the editing scenes.

To evaluate the generalizability of our method in different domains, we further illustrate the results in Fig.6, comparing with HFGI and CLCAE. For artistic portraits, we train our framework in the FFHQ dataset (i.e., human face domain) and only test on those out-of-domain images without any fine-tuning. We can see that our method works better than all other methods in both reconstruction and editing. For cars, we both train and test in the car domain, more details have been included in Appendix. We illustrate two difficult editing scenes in the last two rows of Fig.6, and it shows our method can generate more realistic cars and keep closer background details than other methods. All of these evaluations have demonstrated our framework can work well in various domains without overfitting into a specific domain.

Visualization of residual additions. Our method employs a coarse-to-fine manner to add residual features. A visualization example is illustrated in Fig.7. We can notice that the residual features between stage 1 and stage 2 mainly include coarse details, such as hair and clothes shape. With the development of stages, details become finer. Our framework can effectively align each residual feature with the edited layout, resulting in high-quality editing.

Ablation study. The different effects of each proposed component are compared in Fig.8. By showing the change in the

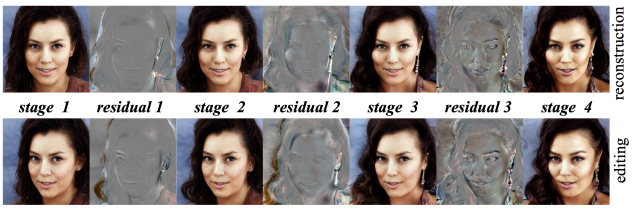

Figure 7: Visualization of our generation results and residuals in different stages of reconstruction and editing.

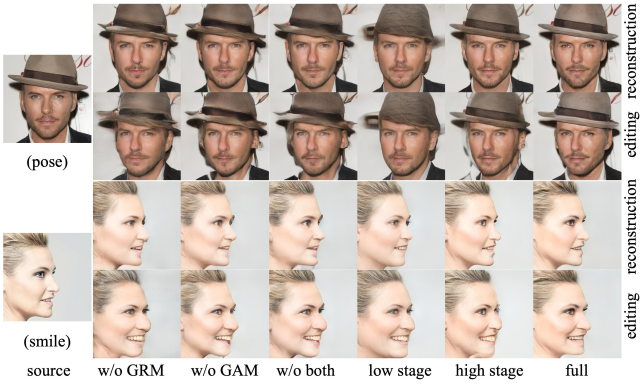

Figure 8: Ablation studies, where low (high) stage indicates only adding residuals in a single low-level (high-level) stage.

image content, we demonstrate our GRM and GAM in the first three columns and our multi-stage addition manner in the 4th and 5th columns. We can observe that worse detail preservation occurs in both reconstruction and editing in *without GRM* (1st column) due to the lack of exact details extracted by GRM. Edited results own heavy artifacts in *without GAM* (2nd column), such as the hat of the man and the mouth of the woman, as there is no details refinement conducted by GAM in editing. Using both modules has the best result (6th column), which demonstrates that our GRM effectively supplements details and GAM correctly suppresses artifacts. Moreover, adding residual features in a single *low-level stage* (i.e., coarser feature map) leads to poor reconstruction quality (4th column), and only adding in a single *high-level stage* (i.e., finer feature map) results in editing artifacts (5th column). Our multi-stage method can achieve an excellent distortion-editability trade-off.

5 Conclusions

In StyleGAN inversion and editing, we propose to gradually add details information for the first time, which achieves a unity of both high-quality detail preservation and high editability. In particular, a novel dual-stream framework is proposed to calculate residual features step by step and then align them with edited images. Further, We utilize a self-supervised training strategy to train both streams simultaneously. Extensive experiments have shown the effectiveness of our framework and the improvement over existing methods in terms of reconstruction and editing.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant U19A2057 and the National Science Fund for Excellent Young Scholars under Grant 62222212.

References

Abdal, R.; Qin, Y.; and Wonka, P. 2019. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF international conference on computer vision*, 4432–4441.

Abdal, R.; Qin, Y.; and Wonka, P. 2020. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8296–8305.

Abdal, R.; Zhu, P.; Mitra, N. J.; and Wonka, P. 2021. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3): 1–21.

Alaluf, Y.; Patashnik, O.; and Cohen-Or, D. 2021. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6711–6720.

Alaluf, Y.; Tov, O.; Mokady, R.; Gal, R.; and Bermano, A. 2022. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, 18511–18521.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

Dinh, T. M.; Tran, A. T.; Nguyen, R.; and Hua, B.-S. 2022. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11389–11398.

Härkönen, E.; Hertzmann, A.; Lehtinen, J.; and Paris, S. 2020. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33: 9841–9850.

Hu, X.; Huang, Q.; Shi, Z.; Li, S.; Gao, C.; Sun, L.; and Li, Q. 2022. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11337–11346.

Huang, Y.; Wang, Y.; Tai, Y.; Liu, X.; Shen, P.; Li, S.; Li, J.; and Huang, F. 2020. Curricularface: adaptive curriculum learning loss for deep face recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5901–5910.

Huh, M.; Zhang, R.; Zhu, J.-Y.; Paris, S.; and Hertzmann, A. 2020. Transforming and projecting images into class-conditional generative networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 17–34. Springer.

Kang, K.; Kim, S.; and Cho, S. 2021. Gan inversion for out-of-range images with geometric transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13941–13949.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020a. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33: 12104–12114.

Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34: 852–863.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020b. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Liu, H.; Song, Y.; and Chen, Q. 2023. Delving StyleGAN Inversion for Image Editing: A Foundation Latent Space Viewpoint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10072–10082.

Mao, X.; Cao, L.; Gnanha, A. T.; Yang, Z.; Li, Q.; and Ji, R. 2022. Cycle encoding of a StyleGAN encoder for improved reconstruction and editability. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2032–2041.

Moon, S.-J.; and Park, G.-M. 2022. Interestyle: Encoding an interest region for robust stylegan inversion. In *European Conference on Computer Vision*, 460–476. Springer.

Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2085–2094.

Pehlivan, H.; Dalva, Y.; and Dundar, A. 2023. Styleres: Transforming the residuals for real image editing with stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1828–1837.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; and Cohen-Or, D. 2021. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2287–2296.

Roich, D.; Mokady, R.; Bermano, A. H.; and Cohen-Or, D. 2022. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 42(1): 1–13.

Shen, Y.; Gu, J.; Tang, X.; and Zhou, B. 2020a. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9243–9252.

Shen, Y.; Yang, C.; Tang, X.; and Zhou, B. 2020b. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 44(4): 2004–2018.

Shen, Y.; and Zhou, B. 2021. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1532–1540.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tov, O.; Alaluf, Y.; Nitzan, Y.; Patashnik, O.; and Cohen-Or, D. 2021. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4): 1–14.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.-P.; Yu, N.; and Fritz, M. 2021. Hijack-gan: Unintended-use of pretrained, black-box gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7872–7881.

Wang, T.; Zhang, Y.; Fan, Y.; Wang, J.; and Chen, Q. 2022. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11379–11388.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wei, T.; Chen, D.; Zhou, W.; Liao, J.; Zhang, W.; Yuan, L.; Hua, G.; and Yu, N. 2022. E2Style: Improve the efficiency and effectiveness of StyleGAN inversion. *IEEE Transactions on Image Processing*, 31: 3267–3280.

Wu, Z.; Lischinski, D.; and Shechtman, E. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12863–12872.

Yao, X.; Newson, A.; Gousseau, Y.; and Hellier, P. 2022. A style-based gan encoder for high fidelity reconstruction of images and videos. In *European conference on computer vision*, 581–597. Springer.

Zhang, M.; Lucas, J.; Ba, J.; and Hinton, G. E. 2019. Lookahead optimizer: k steps forward, 1 step back. *Advances in neural information processing systems*, 32.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, 597–613. Springer.