International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM06926
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Maskfacegan: Masked GAN Latent Code Optimization for High-Resolution Face Editing

[1] Chimata Ravya, [2] Gangineni Lakshmi Naga Chandrika [3] Ajmira Prem Sagar, [4] Nimmagadda Chandra Sekhar

[1,2,3] B. Tech Students, Department of CSE, RVR & JC College of Engineering, Chowdavaram, Guntur, A.P, India.

[4]Assistant Professor, Department of CSE, RVR & JC College of Engineering, Chowdavaram, Guntur, A.P, India,

E-mail: y22cs026@rvrjc.ac.in , Y22cs049@rvrjc.ac.in , y22cs005@rvrjc.ac.in , nimmagadda65@gmail.com

----------***----------

Abstract:

Face editing is a key research area in computer vision, but existing methods often suffer from low-resolution outputs, visual artifacts, and limited control over specific facial attributes, leading to unintended changes. This paper introduces MaskFaceGAN, a novel approach for precise local facial attribute editing. It optimizes the latent code of a pre-trained StyleGAN2 model using multiple constraints to preserve image content, accurately generate desired attributes, and restrict modifications to selected regions. These constraints are guided by a differentiable attribute classifier and face parser. Experimental results on FRGC, SiblingsDB-HQf, and XM2VTS datasets demonstrate that MaskFaceGAN achieves high-quality, high-resolution (1024 × 1024) edits with improved control and reduced attribute entanglement compared to existing methods.

Index Terms— Facial attribute editing, generative adversarial networks, GAN inversion, latent code optimization.

I. Introduction

Face attribute editing focuses on transforming specific visual features of a face image while maintaining realism and identity. With the rapid development of deep learning, especially CNNs and GANs, this field has seen significant improvements in generating high-quality, photo-realistic results. These techniques are widely used in applications such as virtual makeup, aging simulation, identity protection, and digital content creation.

Traditional image-to-image translation methods rely on encoder–decoder architectures, which are fast and efficient but limited in producing high-resolution outputs and often introduce distortions. On the other hand, GAN inversion-based methods leverage powerful pre-trained models like StyleGAN2 to generate highly detailed images. However, these methods typically operate on global latent representations, where multiple facial attributes are entangled, making it difficult to modify one feature without unintentionally affecting others.

The proposed MaskFaceGAN addresses these limitations by enabling precise local editing of facial attributes. It employs a latent space optimization strategy guided by multiple constraints to ensure accurate and controlled modifications. A differentiable attribute classifier enforces the presence or absence of desired attributes, while a face parsing model identifies different facial regions, allowing spatially selective edits. This ensures that only targeted regions are modified, preserving the rest of the image. Additionally, the method incorporates a blending mechanism to maintain the natural appearance and identity of the original face. MaskFaceGAN also supports flexible operations such as adjusting the intensity of attributes, editing multiple attributes simultaneously, and modifying the size or shape of facial components.

Extensive experiments on high-resolution datasets confirm that MaskFaceGAN produces superior results with minimal artifacts and significantly reduced attribute entanglement. Overall, the approach provides

a robust and efficient solution for high-resolution, fine-grained facial

Fig. 1. This paper introduces MaskFaceGAN, a novel approach to face attribute editing capable of generating high-resolution, artefact-free and photo-realistic editing results through a carefully designed optimization procedure applied over the StyleGAN2 latent space. The presented (1024 × 1024) examples show editing results for four target attributes. Best viewed zoomed-in

II. Related Work

In this section, we present prior work closely related to our paper. We discuss Generative Adversarial Networks (GANs), research on pre–trained GANs and face editing techniques.

A. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are one of the most widely used generative models in image processing and computer vision [10]. They are generally categorized into two types: unconditional and conditional GANs. Unconditional GANs generate images purely from random noise without any external guidance, whereas conditional GANs incorporate additional inputs to control the content of the generated images. These inputs can take various forms, such as class labels [18], [19], graph structures [20], [21], object layouts [22], or even textual descriptions [23], [24], [25], allowing greater control over the output. The advancement of GAN-based image generation has largely been driven by improvements in model architectures and training strategies. For instance, DCGAN [26] introduced a convolutional architecture along with key design practices like batch normalization and suitable activation functions for stable training. Later, Karras et al. [27] proposed a progressive training strategy, where models gradually learn to generate higher-resolution images by adding layers over time, leading to high-quality megapixel outputs.

Building on these ideas, StyleGAN [28] introduced a novel design inspired by style transfer techniques. It replaced the traditional latent space with an intermediate latent representation, improving both interpolation and disentanglement of features. It also incorporated adaptive instance normalization and stochastic noise inputs to enhance fine details in generated images. This architecture was further refined in StyleGAN2 [17], which addressed issues like visual artifacts and significantly improved image quality, achieving state-of-the-art performance in unconditional image generation. In addition to architectural innovations, various loss functions and regularization techniques have been proposed to further stabilize training and improve output quality [29], [30], [31], [32], [33]. For a more comprehensive understanding of GAN developments, recent survey papers provide detailed insights into the evolution and applications of these models [34], [35].

B. Studies on Pre-Trained GANs

Training GAN models is computationally expensive. For instance, developing and training StyleGAN2 required approximately 51 Volta GPU years [17]. Due to this high cost, researchers have increasingly focused on leveraging pre-trained GANs for various image generation and manipulation tasks. Several studies have explored the capabilities of pre-trained GANs. Bau et al. [36] demonstrated localized object insertion and removal, while Jahanian et al. [37] studied latent space traversals to achieve simple transformations like brightness and zoom adjustments. Goetschalckx et al.

[38] explored latent space navigation to enhance image memorability, and Yang et al. [39] investigated how layer activations relate to image semantics, enabling control over layout and color schemes. Other notable works include PULSE [40], which applies StyleGAN for face super-resolution, and InsetGAN [41], which combines multiple GANs for tasks like face-conditioned body synthesis.

Similar to these approaches, MaskFaceGAN utilizes a pre-trained GAN model. However, unlike prior work, it uses the GAN primarily as a proxy to generate semantically meaningful content in specific spatial regions. The generated content is then integrated with the original image to produce the final edited output, enabling more controlled and localized edits.

C. Face Editing

A wide range of face editing techniques have been proposed in the literature [1], [2], [42], [43]. Some methods, such as [6] and [44], rely on user-provided sketches to guide the editing process. Others, like [11] and [12], focus on learning disentangled latent representations to better control image generation. Lee et al. [45] introduced Mask GAN, a conditional GAN that enables editing of specific facial components, highlighting the advantages of localized modifications.

Encoder–decoder architectures have produced particularly strong results in face editing. For example, Star GAN [5] uses a cycle-consistent image-to-image translation framework [46] to modify multiple facial attributes. AttGAN [7] improves this approach by replacing cycle consistency with reconstruction constraints, while STGAN [8] further enhances performance through selective transfer units. Although these models generate visually appealing results, they are generally limited to lower resolutions and often produce artifacts in high-resolution scenarios. More recent approaches leverage pre-trained GANs for editing. Abdal et al. [15], [16] demonstrated that images can be embedded into the latent space of StyleGAN and manipulated for tasks like style transfer and face editing. InterFaceGAN [9], [14] introduced a method that modifies images by moving their latent codes along learned linear directions corresponding to specific attributes. While effective, such linear manipulations often lead to attribute entanglement, where altering one feature unintentionally affects others.

MaskFace GAN addresses these limitations by directly optimizing the latent code of an input image using multiple constraints. These constraints not only control the desired semantic changes but also restrict modifications to specific spatial regions. This non-linear optimization process enables precise, localized edits while significantly reducing attribute entanglement, resulting in more accurate and realistic editing outcomes compared to existing methods.

III. Methodology

A. Background and Problem Formulation

The goal of face attribute editing is to manipulate a given input image $I \in \mathrm{R}^{3\times m\times}$ in a photo-realistic manner according to some target semantics aaa, i.e.,

$$\psi_a : I' \rightarrow I \in \mathrm{R}^{3\times m\times n}, \qquad (1)$$

where I′I′I′ is the edited image and the semantics are defined by facial attributes (e.g., “Blond hair”, “Big nose”).

Recent methods implement this mapping using GAN inversion [13], [14]. In this process, the input image III is first embedded into the latent space of a pre-trained GAN GGG, resulting in a latent code www. This code is then modified based on the target attribute, i.e., ., ψa : w '→ w∗, and finally, the edited image I ′ is generated by evaluating w∗ through G, that is, I ′ = G(w∗).

MaskFaceGAN follows this framework but introduces improvements for better control and localized editing.

B. Overview of MaskFaceGAN

MaskFaceGAN is based on a latent code optimization strategy guided by multiple constraints:

- Appearance preservation: Keeps unchanged regions close to the original image.
- Semantic constraint: Ensures correct attribute generation.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM06926
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

• Shape constraint: Controls the exact regions where edits occur.

The method operates in three steps:

Step 1: Initialization
Given an input image III and attribute label a, a face parser S identifies regions to preserve S_{skin} and regions to edit S_{Star}.

Step 2: Latent Code Optimization
The latent code is optimized to produce an intermediate image I_G that reflects the target attribute in S_{Star} while preserving S_{skin}. This is achieved using an attribute classifier C and spatial constraints from S, with losses back propagated into the latent space.

Step 3: Blending
The generated image $I_G$is combined with the original image III to produce the final edited result I′, ensuring natural appearance.

C. Models

MaskFaceGAN relies on three main components:

· The Generator (G) is based on StyleGAN2 [17], a state-of-the-art GAN model specialized for generating photo–realistic facial images. Following established edit-ing methodology [15], [16] the extended latent space[1] W^+ of StyleGAN2 is used for encoding image semantics. As a result, an image is represented through a con-catenation of $n_c = 18$ different 512-dimensional latent
vectors w_i, one for each layer of the model, i.e., $w = G(w, n)$, each classification head C_a predicts the probability of a facial attribute being present in I_G, i.e., $c_k = C_k(I')$ for the k-th attribute.

The Attribute Classifier (C) is designed around the multi–task tree neural network from [47] and consists of several shared layers and K classification heads (leaf branches), one

for each of the K attributes supported (for editing) by MaskFaceGAN. Given an image $I_G = G(w, n)$, each classification head C_a predicts the probability of a facial attribute being present in I_G, i.e., $c_k = C_k(I')$ for the k-th attribute.

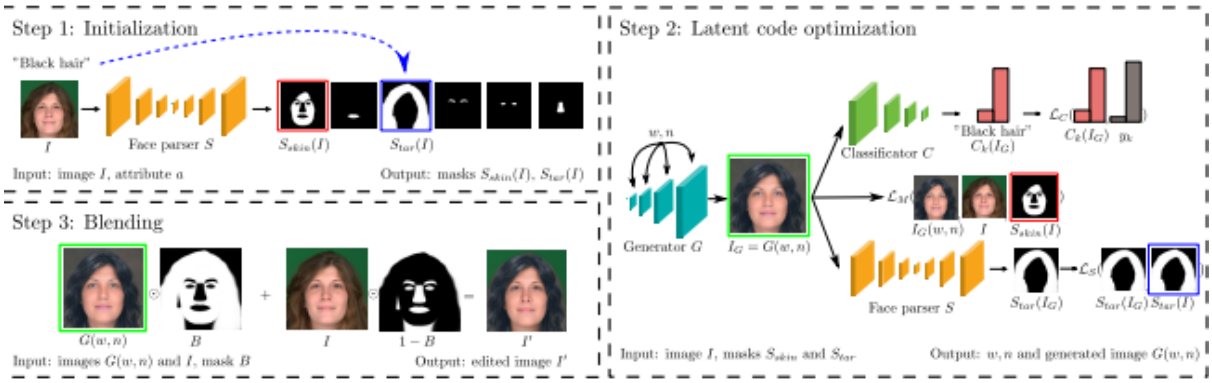

Fig. 2. Overview of MaskFaceGAN, illustrated with the “Black hair” target attribute.

The Face Parser (S) is built around DeepLabV3 [48] and provides pixel-level probability predictions for various face components/regions, as illustrated in Fig. 3. These facial regions are then associated with specific attributes that can be edited within the given region in accordance with Table I. Formally, the model implements a mapping from an image I to a tensor of probabilities along the channel dimension, i.e.: $S : \mathrm{R}^{3\times n\times m} \rightarrow [0, 1]^{L\times n\times m}$, where L is the number of parsed categories (face com-ponents). For MaskFaceGAN, two principal channels are used. The first one is the skin region, $S_{skin} \in [0, 1]^{n\times m}$, which facilitates preservation of facial characteristics unrelated to the change in the targeted semantics.

Fig. 3. Illustration of the ground truth used to train the face parser *S*. The presented regions are associated with specific (local) attributes that can be edited, as summarized in Table I

Supported Attributes & Embeddings: MaskFaceGAN is based on the idea that many facial attributes are localized, meaning they affect only specific regions of the face. This assumption helps reduce the common problem of attribute entanglement, where modifying one feature unintentionally alters others. The model focuses on 14 facial attributes that are naturally linked to particular facial regions (as listed in Table I). However, with minor modifications, the method can also handle more global attributes such as age and gender, as discussed in Section -F.

A key advantage of MaskFaceGAN is its ability to perform local embedding. Instead of embedding the entire image into the latent space, it selectively embeds only the relevant regions associated with the target attribute. For instance, when editing the "Pointy nose" attribute, only the nose region and its surrounding area are embedded and optimized. This ensures that the desired changes are applied strictly within the targeted region, without affecting other parts of the face.This localized optimization is achieved through carefully designed constraints, which guide the latent code to generate the intended attribute while preserving the rest of the image. As a result, MaskFaceGAN enables precise, region-specific editing with improved control and minimal unintended modifications.

B. Latent Code Optimization

1) Appearance Preservation: The goal of facial attribute editing is to alter specific (targeted) image semantics, while preserving all (or most) other visual characteristics of the input images. To ensure that image regions not associ-ated with the targeted attributes are preserved, a (local) appearance–preservation constraint is used during optimiza-tion. The constraint is defined as a masked mean squared error (MMSE)

2) Semantic Content: The constraint in Eq. forces certain image regions in IG to be preserved w.r.t. I , while the rest is allowed to change. MaskFaceGAN, thus, synthesizes the remaining image pixels in accordance with the targeted semantics by considering a semantic–content constraint in the optimization procedure. The constraint ensures that the latent code w produces an image IG with the desired facial attributes and is defined as the average Kullback–Leibler (KL) divergence DK L between the smoothed ground truth proba-bility distribution and classifier predictions for the targeted attribute(s) [49], i.e.:

$$LC = \frac{1}{K} \sum_{k=1}^{K} DKL(C_k(IG), y_k),$$

where K denotes the number of targeted facial attributes, Ck stands for the attribute classifier prediction corresponding to the k-th attribute and $y_k \in \{\epsilon, 1-\epsilon\}$ is the smoothed ground truth that denotes the absence or the presence of the desired attribute, respectively. The value of ϵ can be used to set the intensity of the desired attribute, e.g., the intensity of lipstick presence when editing the "Wearing lipstick" attribute.

2) Target Region Shape: Because image content with the targeted semantics is first synthesized by MaskFaceGAN and later blended with the original image, it is critical that the shape of the targeted facial

regions is preserved. To this end, the proposed approach constrains the shape of the targeted region with the help of the face parser S using: LS = ||Star (I) − Star (IG)||2, here Star is again a probabilistic mask of the spatial region encourages the generated image IG and the input image I to be as similar as possible within the area defined by Sskin.

4) Component Size: While the main goal of MaskFaceGAN is to produce convincing manipulations of existing image content, additional components can be incorporated into the framework to enable further editing capabilities. Specifically, region we introduce a scaling factor α and integrate it into an objective that considers the component size when optimizing for the latent code w. The objective is defined as the KL divergence between the initial component portion star (I) and the desired portion star (IG) in the generated image IG, i.e.:

LP = DK L (star (IG), αstar (I))

We note that this term is optional and can be excluded from the optimization procedure by setting the corresponding weighting factor to 0 - see final objective in Eq. (9) for details.

5) Flexible Spatial Constraints:

The appearance–preservation and target–shape optimization constraints, defined in Eqs. (2) and (4), impose significant limitations on the spatial regions associated with the targeted facial attributes. The appearance–preservation constraint does not allow to grow relevant facial components if they overlap with the skin region. Similarly, the target–shape objective forces the edited image to have exactly the same target component shape as the original image, which in some cases might not be desired. For example, when editing hair color, the target shape needs to be preserved, but when editing hair shape (e.g., "Straight hair" or "Wavy hair") modifications of the target image region must be allowed.

To deal with such issues, we relax the optimization constraints from Eqs. (2) and (4) and incorporate mechanisms into MaskFaceGAN that allow for more flexible spatial editing. Specifically, in each iteration of the optimization procedure, we first compute the target region on the generated image Star (IG). Next, we subtract this region from Sskin(I) for the appearance–preservation constraint to preserve less pixels. For the target–shape objective, the region is added to the target region of the original image Star (I ′) to allow region growth. Here, we also require that the combined region covers at least the original component shape to avoid visual artefacts. The final (relaxed) appearance–preservation LM and target–shape LS constraint used by MaskFaceGAN are, hence, defined as:

LM = || min(Sskin(I) − Star (IG), 0) ⊙ (IG − I)||2, and (7)

LS = || max(Star (I) + Star (IG), 1) − Star (IG)||2,

where min and max denote pixel–wise minimum and max-imum operations. The impact of these constraints on the appearance of a few sample images is shown in Fig. 4.

TABLE I

ATTRIBUTES SUPPORTED FOR EDITING BY MASKFACEGAN AND CORRESPONDING FACIAL AREAS MANIPULATED BY THE PROPOSED APPROACH TO ENFORCE DESIRED SEMANTICS

Face attribute (for editing)	Face region (returned by S)
Blond, Brown, Black, Grey, Straight, and Wavy hair	Hair
Wearing lipstick, Smiling, Mouth slightly open	Lower and Upper lip, Mouth
Bushy eyebrows, Arched eyebrows	Left and Right eyebrow
Pointy nose, Big nose	Nose
Narrow eyes	Left eye, Right eye

C. Noise Component Optimization

While the semantic content of the edited images is controlled by the latent code w, the high-frequency facial details that ensure photo realism are defined by the noise components n. After the latent code w is optimized, w is frozen and

MaskFaceGAN proceeds to optimize n, similarly to [16]. Two key issues are considered during optimization, i.e.:

· Adversarial solutions: Due to the high-dimensionality of n, a naive optimization procedure based on Eq. (10) can lead to editing results akin to adversarial examples, i.e., generated images that satisfy all constraints but do not exhibit the desired semantics. To avoid such settings the objectives related to semantics and target–region shape are not considered when optimizing for n.

· Overfitting: The optimization procedure can lead to solu-tions that perfectly reproduce all stochastic details of the original face (e.g., freckles, wrinkles) except for facial areas altered by the editing procedure. This mismatch between preserved and altered image regions results in unnatural appearances and a "copy-paste" look. To avoid such overfitting and ensure a reasonable amount of details in the preserved as well as generated image regions, a noise regularization term is used, similarly to [17].

Based on the above considerations, MaskFaceGAN's noise-related optimization objective takes the following form:

$$L_n = \lambda_M L_M + \lambda_R \quad L_{i,j}, \quad (10)$$

The goal of this regularization term is to ensure that the noise components follows a normal Gaussian probability distribution by preserving the mean and standard deviations of neigh-bouring values. At every step of the optimization, each noise component larger than 8×8 is downsampled in a pyramid-like fashion to a resolution of 8×8 by averaging 2×2 neighbouring values. In the above equation, $n_{i,j}$, thus, denotes the i -th noise component at the original resolution (j = 0) or a given level of the downsampling pyramid (j > 0). The number of elements of $n_{i,j}$ is denoted as $|n_{i,j}|$ and the corresponding regularization term as $L_{i,j}$. The impact of the term is illustrated in Fig. 5.

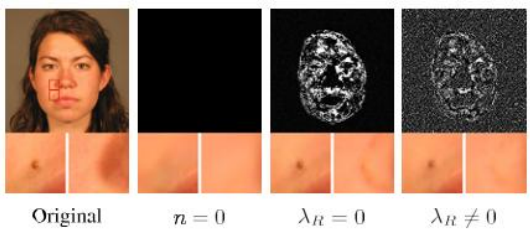

Fig. 5. Embedding quality with respect to different noise optimization settings when editing a nose-region attribute.

The first column shows the original image and a closeup of two face regions. Without noise optimization high-frequency details are not embedded, but smooth transitions are generated between the skin and the nose regions (second columns). Optimizing for the noise directly perfectly embeds fine skin details, but does not produce smooth transitions (third column), while th regularized optimization generates a reasonable trade–off between details and transitions (last column).

A. *Blending*

In the final step, MaskFaceGAN blends the image generated based on the optimized latent code w and noise components n, $I_G = G(w, n)$, with image regions in the input image I that were not considered during optimization. These regions correspond to the background and non-edited facial compo-

nents. To facilitate this step, a blending mask is computed as $B = S_{skin}(I) + S_{tar}(I)$ for most attributes and $B = S_{skin}(I) + S_{tar}(I_G)$ when editing hair shape to account for potential modification of the hair region. The final image I' is generated as

$$I' = B \odot I_G + (1 - B) \odot I. \quad (12)$$

The blending step is visualized in the bottom left of Fig. 2. Note that the blending operation considers the skin region from the optimized image I_G. This is needed to allow Mask-FaceGAN to also change the size and shape of the targeted attributes in a visually convincing manner, with smooth tran-sitions and without visual artefacts where λ_M and λ_R are again weighting factors and the noise regularization term $L_{i,j}$ is defined as:

$$L_{i,j} = \left(\frac{1}{|n_{i,j}|} \sum_{x,y} n_{i,j}(x, y) \cdot n_{i,j}(x-1, y) \right)^2 + \left(\frac{1}{|n_{i,j}|} \sum_{x,y} n_{i,j}(x, y) \cdot n_{i,j}(x, y-1) \right)^2 . \quad (11)$$

EXPERIMENTAL SETUP

A. *Datasets and Experimental Splits*

Four high-resolution images datasets are used in the experi-ment with MaskFaceGAN, i.e., Flickr-Faces-HQ (FFHQ) [17], FRGC [52], XM2VTS [53], and SiblingsDB–HQf [51]. The

training and testing datasets were carefully selected based on their image licences and the presence of individual consent towards the use of personal face images towards research purposes, as well as technical criteria, such as dataset size, image resolution, image quality, and available

annotation. A brief summary of the datasets and experimental splits used is

given below:

· Flickr-Faces-HQ (FFHQ) [17] contains 70, 000 high quality face images at a resolution of 1024 × 1024 pixels. The faces contain considerable variation in terms of age, ethnicity and image background. The images come with Creative Commons licence and were made available from the authors of StyleGAN2. FFHQ is used as the primary training dataset, used to train the generator model (G), the attribute model (C), and the segmentation model (S). We generate binary attribute annotations by following MAAD-Face distillation protocol [54].

TABLE II

HIGH–LEVEL SUMMARY OF THE DATASET AND EXPERIMENTAL SETUP USED WITH MASKFACEGAN. NOTE THAT DATASETS WITH DIFFERENT CHARACTERISTICS AND DIVERSE FACE IMAGES WERE SELECTED FOR THE EXPERIMENTS TO DEMONSTRATE THE MERITS OF THE PROPOSED

Dataset	Image Resolution	Purpose	#Training Images†	#Test Images	Variability Sources
FFHQ [17]	1024 × 1024	Training of G, C, S	70, 000	n/a	Age, ethnicity, background
FRGC [50]	1704 × 2272	Testing	n/a	200	Age, ethnicity, gender
XM2VTS [50]	720 × 576	Testing	n/a	200	Age, gender
SiblingsDB–HQf [51]	4256 × 2832	Testing	n/a	200	Age, gender

† The number of training images reported includes both training and validation data.

· Face Recognition Grand Challenge (FRGC) [52] con-sists of high resolution images and 3D scans. The imagery was shot under both controlled and unstructured illu-mination. The image sizes are either 1704 × 2272 or 1200 × 1600 pixels. SiblingsDB-HQf [51] contains frontal, expressionless images of 184 subjects – 92 sibling pairs with a resolution of 4256 × 2832. Images in this dataset were captured in front of a uniform background and under controlled lightning. XM2VTSDB (XM2VTS) [53] consists of digital video recordings of 295 subjects, taken at one month inter-vals over a period of five months. The images were taken under controlled lightning conditions with uniform background. We select 200 high-quality images from each of the testing dataset, i.e. FRGC, SiblingsDB-HQf, and XM2VTS, for the experiments. The images are then processed with a standard face-processing pipeline [27] that involves cropping the face region and resampling to 1024×1024 pixels. This resolution is used in all MaskFaceGAN editing experiments. A high–level

B. *Implementation Details*

The models used by MaskFaceGAN are implemented with publicly available source code. Further details are given below.

· The Generator (G) of MaskFaceGAN is implemented using the official StyleGAN2 release [17] to foster repro-ducibility and ensure a fair comparison with competing techniques from the literature designed around this model.

· The Attribute Classifier (C) is implemented based on the model from [47] and trained on the attribute-annotated FFHQ dataset, which was split into a training, validation, and test split. The training is done for 26 epochs with weighted binary cross entropy to account for the class imbalance in the training data. The learning rate is initially set to 0.05 and decayed to 0.005 on the 40, 000–th training step. The model is optimized with the Nesterov momentum algorithm [55] using a batch size of 32. Augmentations with random horizontal flipping and affine transformations are used to avoid over-fitting.

· The Face Parser (S) is based on DeepLabv3 [48]. It is trained based on the DatasetGAN [56] proto-col for generating synthetic face-parsing ground truth. We use the ground truth provided by DatasetGAN for the facial parsing task that includes 34 facial categories.

· These categories are merged to produce the follow-ing 9 facial regions/classes: “hair”, “skin”, “eyebrows”, “nose”, “eyes”, “mouth”, “neck & clothes”, “earrings”, and “ears”. We generate 100, 000 synthetic facial images and their corresponding ground truth segmentation masks.

epochs using the Adam optimizer [57] with a learning rate of $3 \cdot 10^{-4}$ and a batch size of 24. Data augmentation includes horizontal flipping and affine transformations.

The latent code optimization procedure is conducted in several stages. First, w is initialized with the mean latent code $\bar{w}$, computed from 50, 000 sampled codes $w \in W^+$. Next, it is optimized w.r.t. Eq. (2), so it approximately corresponds to the target face. This initial step was found to be important for the visual quality of the edited images. Finally, the remaining terms are added to enforce semantic and spatial constraints. Similarly to [16], the noise component n is set to 0 and kept constant while optimizing w. Once w converges, it is frozen and the noise component n is optimized independently of w. To identify parameter values that yield visually pleasing editing results, hyper-parameter optimization is used, resulting in weighting factors of $\lambda_M = 2$, $\lambda_C = 0.005$, $\lambda_S = 0.5$, $\lambda_R =$

1. We set $\lambda_S = 0$ for operations where no hair editing isdone. The default value for Eq. (3) is set to $\epsilon = 0.05$. The Adam algorithm [57] is again for the optimization process. The learning rate is set to 0.001 for the latent code w and to 0.1 for the noise component n. Additional implementation details can be found in the publicly released code of MaskFaceGAN.

C. Methods

1) Visual Analysis: We first demonstrate the capabilities of MaskFaceGAN for the task of single attribute editing and include the 14 binary attributes from Table I in the analysis. Images from different datasets are used for the experiments to explore the generalization capabilities of the evaluated approaches across various data distributions. If a face already exhibits a given attribute (e.g., a face wearing lipstick), we generate *inverted* attributes, (i.e., a face without lipstick). Fig. 6 compares editing results produced by MaskFace-GAN and the five competing models on a couple of sample images from FRGC and XM2VTS. Note that 10 attributes are considered per example image to ensure a reasonable image size for the presentation. Among the encoder–decoder models, StarGAN generates the highest amount of visual artefacts. AttGAN and STGAN produce more convincing results, but still induce a certain amount of visual artefacts. These can, for example, be seen with the "Blond hair" attribute in the Fig. 6. The artefacts generated by the encoder-decoder methods stem from difficulties in balancing multiple loss terms commonly
used when training such methods.

InterFaceGAN and InterFaceGAN–D are most closely related to MaskFaceGAN and achieve higher–quality editing result than the encoder–decoder models due to the

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM06926
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Fig. 6. Comparison of MaskFaceGAN and five state–of–the–art attribute editing models from the literature. Editing results are presented for 14 distinct facial attributes with spatial correspondences. For attributes already present in the image, editing inverts the result (e.g., removes the lipstick for "wearing lipstick" if it is already there) - displayed in italic. Results on the top correspond to a sample image from FRGC and results at the bottom to an image from XM2VTS. Best viewed zoom in.

use of the StyleGAN2 generator. The vanilla version of InterFaceGAN yields convincing target semantics, but due to the information entanglement in the latent codes, often changes correlated attributes in the process. This is best seen with the "Grey hair" attribute in Fig. 6, where the edited faces appear much older than the originals. InterFaceGAN–D is able to

remove some of this entanglement, but this requires a manual analysis of attribute correlations to exclude unwanted facial semantics from the editing procedure. We also observe an interesting behavior with the InterFaceGAN models, in that the same hyper–parameter setting (i.e., the magnitude of the latent code movement), results in attribute changes of different intensity for images of different characteristics – see, for example, the "Blond hair" results in Fig. 6.

Compared to the competing models, the proposed Mask-FaceGAN approach: *(i)* exhibits better disentanglement characteristics due to the latent space optimization procedure, which relies on semantic and spatial constraints, *(ii)* ensures artefact–free high–resolution attribute editing with convincing image semantics, *(iii)* preserves important image details (e.g., facial areas not related to the target attribute or background), and *(iv)* does not require manual hyper–parameter tuning for each probe image separately.

2) Quantitative Evaluation: To evaluate attribute editing performance in a quantitative manner, prior works [7] and [8] reported a measure quantifying attribute generation accuracy. Because MaskFaceGAN tries to maximize this exact measure during latent code optimization, we use an alternative approach to ensure a fair comparison. Specifically,

Fréchet Inception Distances (FID) to quantify performance and then present results of a user study, similarly to [8].

· FID Score Analysis. The Fréchet Inception Distance (FID) [58] represents a common measure of image quality, predominantly used in the evaluation of GANs. We report FID scores for each dataset considered in our evaluation by first generating attribute specific FID scores and then averaging over all attributes. The facial images are rescaled to 299 × 299 pixels before extracting features. Table III shows that MaskFaceGAN achieves the lowest FID scores on all three test datasets, significantly outperforming all five competing editing models. The lower scores can mostly be attributed to the high quality

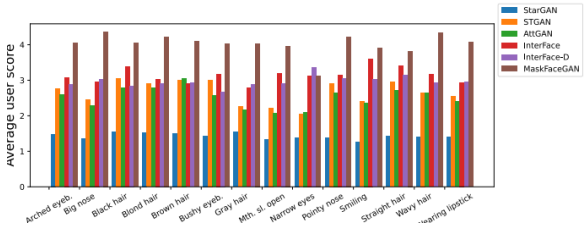

Fig. 7. Comparison of user study scores, averaged across all three dataset for individual attributes. As can be seen, MaskFaceGAN achieves highly competitive results with all targeted attributes. The figure is best viewed in color.

TABLE III

FID SCORES PRODUCED BY THE EVALUATED MODELS ON THREE EXPERIMENTAL DATASETS (LOWER IS BETTER)

Method	FRGC	SiblingsDB–HQf	XM2VTS
StarGAN [5]	189.82	209.38	160.80
AttGAN [7]	53.63	47.27	40.47
STGAN [8]	43.12	44.24	29.81
InterFaceGAN [9], [14]	41.35	46.48	44.48
InterFaceGAN–D [9], [14]	39.49	45.39	40.34
MaskFaceGAN (ours)	**19.96**	**17.38**	**20.81**

TABLE IV

USER STUDY RESULTS, WHERE HUMAN RATERS WERE SHOWN EDITING RESULTS OF ALL TESTED MODELS AND ASKED TO SELECT THE BEST ONE. REPORTED IS THE FRACTION OF TIMES [IN %] A MODEL WAS CHOSEN AS THE OVERALL BEST (HIGHER IS BETTER)

Method	FRGC	SiblingsDB–HQf	XM2VTS
StarGAN [5]	2.97 %	4.16 %	0.57 %
AttGAN [7]	6.27 %	7.01 %	8.24 %
STGAN [8]	14.69 %	9.93 %	7.57 %
InterFaceGAN [9], [14]	9.24 %	21.09 %	22.32 %
InterFaceGAN–D [9], [14]	11.22 %	12.04 %	13.31 %
MaskFaceGAN (ours)	**55.61 %**	**45.77 %**	**47.99 %**

TABLE V

USER STUDY RESULTS, WHERE HUMAN RATERS WERE ASKED TO RATE THE QUALITY OF THE EDITED IMAGES ON A 5-POINT LICKERT SCALE (HIGHER IS BETTER). REPORTED IS THE AVERAGE SCORE AND CORRESPONDING STANDARD DEVIATION

Method	FRGC	SiblingsDB–HQf	XM2VTS
StarGAN [5]	1.43 ± 0.89	1.52 ± 0.91	1.31 ± 0.65
AttGAN [7]	2.82 ± 1.04	2.46 ± 0.92	2.42 ± 1.07
STGAN [8]	3.01 ± 1.21	2.65 ± 1.02	2.47 ± 1.12
InterFaceGAN [9], [14]	3.00 ± 1.05	3.26 ± 1.04	3.07 ± 1.20
InterFaceGAN–D [9], [14]	2.97 ± 1.15	3.11 ± 1.09	2.78 ± 1.23
MaskFaceGAN (ours)	$\mathbf{4.20 \pm 1.31}$	$\mathbf{4.01 \pm 1.22}$	$\mathbf{3.94 \pm 1.32}$

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM06926
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

· User Study. Following [8], we conduct a user study using a crowdsourcing platform to analyze the quality of the edited images. Here, the users (raters) were shown edited images of all considered models and asked to select the most convincing one based on the following instructions: *"Choose the image that changes the attribute more suc-cessfully, is of higher image quality and better preserves the identity and fine details of the source image."*. Addi-tionally, they were also instructed to rate images on a 5–point Likert scale, where a higher number represents better image quality. A single user study covered all test images from a given dataset and was performed over all 14 attributes. Images were shown in random order for a fair comparison. The results, reported in Tables IV and V, show that MaskFaceGAN was most frequently selected as the best among the evaluated techniques and also received the highest average scores (on the 5–point Likert scale) on all three dataset. These observations are further supported

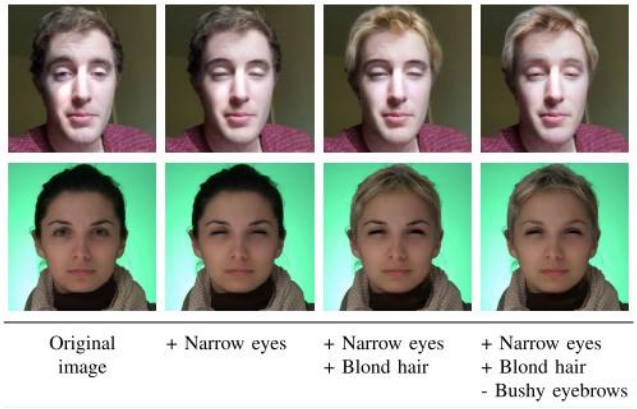

Fig. 8. Editing multiple attributes with MaskFaceGAN. Every image is the result of a separate optimization procedure and is generated independently from all others.

by the results in Fig. 7, where user scores are reported for each edited attribute separately. The reported results speak of the excellent performance of MaskFaceGAN and competitiveness with respect to existing models.

B. *Characteristics of MaskFaceGAN*

MaskFaceGAN exhibits several desirable characteristics, such as the capability to *(i)* edit multiple attributes with a single optimization procedure, *(ii)* control the intensity of edited attribute, and *(iii)* modify the size of the edited region. We illustrate these characteristics through several visual examples.

1) Multiple Attribute Editing: By averaging the KL diver-gence of the semantic constraint over multiple attributes, MaskFaceGAN can edit multiple binary attributes through a single optimization procedure. Examples of such editing results are presented in Fig. 8 for different numbers of attributes, i.e., $K = \{1, 2, 3\}$. Two interesting observations can be made here: *(i)* even when multiple attributes are edited, the results are still visually convincing and artefact-free, and *(ii)* the joint optimization of several attributes retains considerable correspondence with the original image.

The multiple-attribute editing capabilities of MaskFaceGAN are especially useful when editing hair shape. Because the model is not explicitly aware of characteristics of the original facial region considered during the editing process, it can in a limited number of cases also alter some additional attributes in addition to the targeted attribute, e.g., change the hair

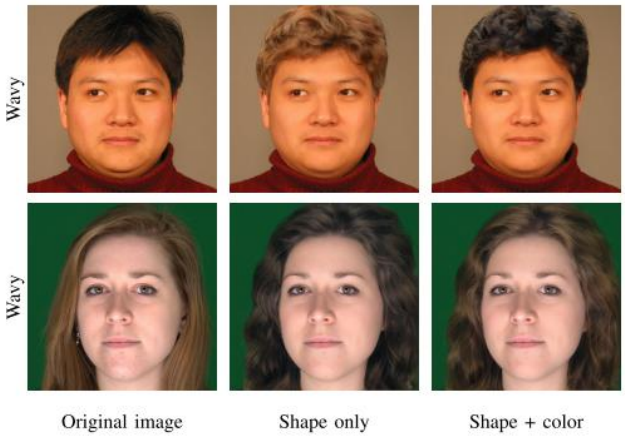

Fig. 9. Editing hair shape with MaskFaceGAN. Editing only the hair shape can lead to changes in hair color (second column). Adding hair color (from the input image) as an additional optimization constraint helps towards preserving the input hair color semantics.color when hair shape edits are targeted. While this may not be desired, MaskFaceGAN can address such problems by defining multiple target attributes. For example, when editing hair shape, all other hair-related characteristics considered by the attribute classifier C can be set to the same target value as in the input image. In Fig. 9, we illustrate this characteristic on a couple of sample images. Observe how the inclusion of hair color in the optimization procedure helps to better

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI:10.55041/ISJEM06926
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

preserve the initial image properties when editing hair shape.

2) Attribute Intensity Control: MaskFaceGAN's semantic constraint is defined by the KL divergence between the pre-dictions of the attribute classifier (C) and the corresponding ground truth. Because the ground truth is smoothed and for a given attribute consists of $y \in \{\epsilon, 1-\epsilon\}$, varying the smoothing parameter ϵ affects the strength (or intensity) of the targeted attribute in the edited images. A few illustrative examples of the impact of ϵ are presented in Fig. 10. As can be seen, MaskFaceGAN allows for fine-grained control over the attribute intensity in the edited images, although the generated variations may not necessarily be smooth w.r.t. the visual appearance change. The results depend on the trained classifier and what it considers an attribute presence with $1-\epsilon$ probability.

3) Component Size Manipulation: MaskFaceGAN can be adapted to include additional constraints. An example con-straint is the desired size of the facial component being manipulated. This is done by including the size manipulation objective from Eq. (6) in the overall optimization objectivein Eq. (9). In Fig. 11 we display results when specifying the portions of the original component size for $\alpha = \{0.5, 1.0, 1.5\}$. The presented examples show MaskFaceGAN's capability to change the size of facial attributes in a photo realistic manner.

4) Combining Editing Constraints: Multiple attribute edit-ing, intensity control and component size manipulation can also be used simultaneously to change several aspects of the input face image with a single application of MaskFace-GAN. Fig. 12 presents an example, where various aspects of the "Wearing lipstick" and "Bushy eyebrows" attributes are manipulated. Note that despite considerable changes to different facial attributes, the results still appear visually convincing.

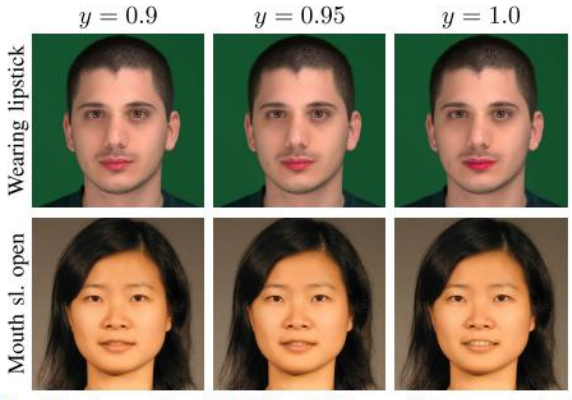

Fig. 10. Visual examples of MaskFaceGAN's capability to control attribute intensity during editing.

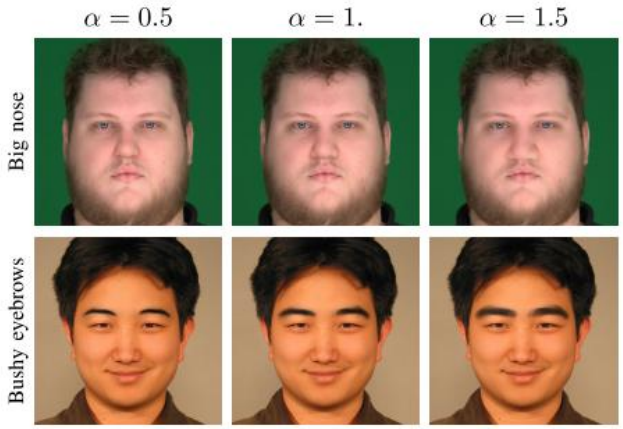

Fig. 11. Visual examples of MaskFaceGAN's capability to control the size of the edited facial components.

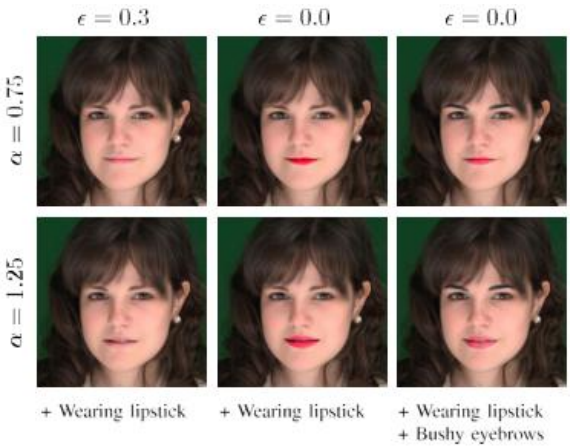

Fig. 12. An example of simultaneous component size manipulation and attribute intensity control. Results are shown for editing a single (i.e., "Wearing lipstick") or two attributes (i.e., "Wearing lipstick" and "Bushy eyebrows") at the same time.

C. Ablation study

To evaluate the impact of different MaskFaceGAN com-ponents on the editing quality, we perform an ablation study on FRGC. Specifically, we focus on two major components: *(i)* the shape constraint from Eq. (4), and *(ii)* the noise optimization procedure. We note that the shape term only affects the hair region and does not impact other attributes.

1) Qualitative Analysis: Fig. 13 demonstrates the effect of different MaskFaceGAN settings. When the noise component is not optimized ($n = 0$), the

edited images contain low frequency image areas, which is most apparent in the hair

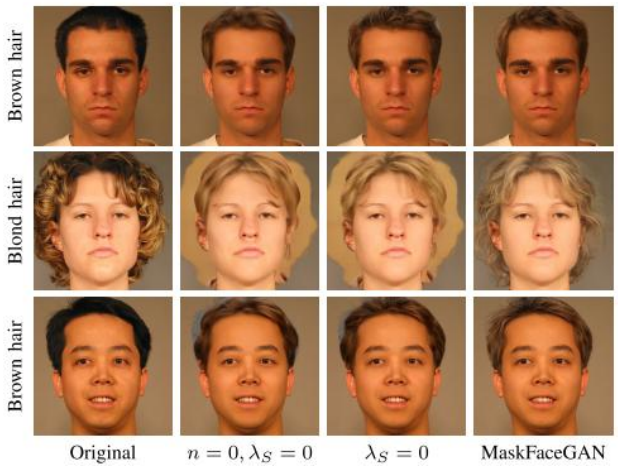

Fig. 13. Qualitative results of the ablation study. The figure shows (left to right): (i) original images, (ii) results without noise optimization and without the shape constraint, (iii) effect of the noise optimization, and (iv) results with noise optimization and the shape term enabled.

TABLE VI
QUANTITATIVE RESULTS OF THE ABLATION STUDY. FID SCORES COMPUTED ON FRGC AVERAGED OVER ALL ATTRIBUTES ARE REPORTED (LOWER IS BETTER)

MaskFaceGAN variant	FID
Local latent code optimization, no shape term	58.85
+ noise optimization	22.78
+ shape term (complete MaskFaceGAN)	**19.96**

region, as shown in the second column of Fig. 13. Similarly, the skin region is also missing details, e.g., beauty marks. The noise optimization ensures that such facial details are present in the image, as can be seen in the third column of Fig. 13.

The absence of the shape term ($\lambda_S = 0$) results in suboptimal blending when dealing with hair modifications. In such settings, the background synthesized by the generator (G) is

blended with the original background, resulting in unconvinc-ing results with visible artefacts. The optimization of this term assures that the generator model considers information about the shape of the hair region during the synthesis step and produces photo realistic editing outputs.

2) Quantitative Analysis: For a quantitative analysis of the ablation results, we report in Table VI mean FID scores computed over the test images of FRGC dataset and averaged over all attributes. Interestingly, the largest gain is obtained by the noise optimization procedure. Enabling the shape term to ensure blending consistency results in additional FID gains. We hypothesize that these gains are a consequence of visually more convincing images, as shown in Fig. 13.

D. Component Analysis

The optimization procedure designed for MaskFaceGAN depends on the utilized attribute classifier (C) and face parser

(S). In this section, we analyze the impact of these two components on the editing results.

1) Attribute Classifier: To explore the impact of the attribute classifier C, we implement 5 additional models in addition to the tree-like classifier from [47] that is used in the original MaskFaceGAN design: two attribute classifiers based on the VGG architecture [59], i.e., VGG16 and VGG19, and

TABLE VII
AVERAGE PREDICTION ACCURACY ON FFHQ FOR THE ATTRIBUTE CLASSIFIERS SELECTED FOR THE COMPONENT ANALYSIS

Model	VGG16	VGG19	ResNet34	ResNet50	ResNet101	Ours
Accuracy	93.6%	93.2%	93.8%	93.9%	93.9%	96.3%

Fig. 14. Impact of different attribute classifiers on the generated edits. For all results, the smoothing factor is set to $\epsilon = 0.05$. The tree-based attribute classifier used in MaskFaceGAN [47] leads to more expressive semantics in the final image compared to other attribute classification models.

three ResNet-based [60] models, i.e., ResNet34, ResNet50 and ResNet101. These models come with different designs and complexity (in terms of parameters). We modify the models to predict multiple attributes by constructing a dedicated fully connected layer at the top with one output for each attribute. The models are trained on the FFHQ dataset using the same optimization procedure and learning rate schedule as utilized with the original tree-like classifier [47]. Table VII shows the classification accuracy (averaged across all attributes) on the FFHQ test split achieved by the different models. We observe that despite differences in design and complexity most models weigh in at a classification accuracy around 93%, whereas the tree-like classifier performs slightly better.

In Fig. 14 we investigate how the characteristics of the attribute classifiers impact results. As can be seen, all clas-sifiers lead to semantically meaningful editing outputs and, even though their performance is close, they produce slight variations in the appearance of the targeted attributes - see, for example, the mouth region in the bottom row of Fig. 14. This observation can be attributed to the different model topologies and differences in the gradients produced during the optimization procedure. Furthermore, it can be noted that the results generated with the tree-based attribute classifier (marked Ours) contain the most expressive semantic content with the most pronounced targeted attributes. We ascribe this fact to the superior classification performance of our attribute classifier, which consequently provides better gradient information compared to the competing models.

Face Parser: MaskFaceGAN uses DeepLabv3 [48] to implement the face parser (S). In this section, we analyze editing results produced with 3 other parsers. We select the following competing models for the comparison: UNet [61] and FCN [62] with two different backbones, i.e., FCN–ResNet50 and FCN–ResNet101, and train them on FFHQ using the same procedure/protocol and optimization method as with DeepLabv3. The performance in terms of the average Intersection over Union (IoU) [63] over the test split of the data is reported in Table VIII. Note that the average IoU scores (in %) are close and vary from around 80% to up to 83%.

TABLE VIII
AVERAGE INTERSECTION OVER UNION (IOU) [IN %] OVER THE FFHQ TEST DATA FOR THE SELECTED FACE PARSERS

Model	UNet	FCN–ResNet50	FCN–ResNet101	DeepLabv3 (ours)
IoU	80.69%	81.95%	80.95%	83.19%

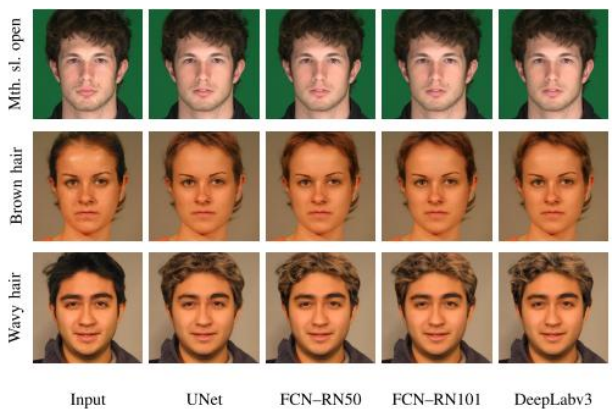

Fig. 15. Impact of different face parsers on the generated edits. The impact on most attributes is negligible, as illustrated, for example, by the "Mouth slightly open" edit in the first row. When the target-region shape loss term is used (hair edits in the second and third row), minute differences can be observed in the generated images.

E. Optimization Procedure and Blending

In this section, we study the optimization procedure and blending step used in MaskFaceGAN to provide better insight into their behavior and impact on the final results.

Optimization Procedure: A two-step process is used in MaskFaceGAN to find the optimal latent representation

F. Global Editing

While the primary purpose of MaskFaceGAN is editing of local attributes that correspond to specific facial regions, the approach can also be extended towards editing of global facial characteristics. The proposed modification allows MaskFace-GAN to edit other attributes, e.g., skin tone, gender, and age. To facilitate global editing, we relax the original appearance-preservation constraint from Eq. (2), so it allows for global image changes. Specifically, instead of using the MMSE-based constraint over the skin area to preserve the initial image appearance, we introduce a perceptual loss over the whole face region, as illustrated in Fig. 18.

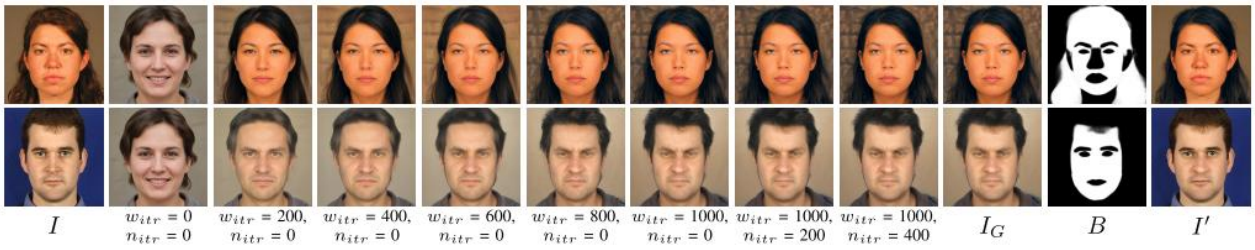

Fig. 16. Evolution of the intermediate G(w, n) results throughout the optimization procedure. The leftmost image shows the input I. The next examplesshow the intermediate results for $G(w, n)$, where the numbers at the bottom correspond to the current optimization iteration of w and n, that is, $witr$ and $nitr$, respectively. The initial image ($witr = 0$, $nitr = 0$) is always the same. During the w latent code optimization procedure, the face shape and target-attribute appearance is adjusted in accordance with the considered constraints. After convergence, the noise component is optimized to introduce realistic fine image details. The final optimization result IG is then blended based on B to generate the output image I'. The attributes edited in the presented examples are "Black hair" (top) and "Big nose" (bottom).

F. Limitations

The results presented so far show that MaskFaceGAN generates competitive (high–quality) editing results when compared to state–of–the–art models from the literature. Nev-ertheless, the approach still exhibits a number of limitations. MaskFaceGAN is based on gradient optimization that takes between 2 and 5 minutes per image on a GeForce GTX 1080. In comparison with encoder–decoder methods that are capable of editing images in milliseconds, the proposed approach is slower by orders of magnitude. However, when compared to related methods, e.g., InterFaceGAN [14], the local embedding

I. CONCLUSION

In this paper, we introduced MaskFaceGAN, a novel approach to high–resolution face image editing. At the core of the approach is a GAN latent code optimization procedure that generates targeted image regions in accordance with spatial and semantic constraints, enforced by pre–trained face parsing and classification networks. Through rigorous experiments on three face datasets, MaskFaceGAN was shown to convincingly alter a wide variety of facial attributes and ensure com-petitive performance when compared to the state-of-the-art. Additionally, the approach was demonstrated to enable unique editing characteristics, including attribute intensity control and component size manipulation.

REFERENCES

[1] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and
J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Inf. Fusion*, vol. 64, pp. 131–148, Dec. 2020.

[2] S. Jiang, Z. Tao, and Y. Fu, "Geometrically editable face image trans-lation with adversarial networks," *IEEE Trans. Image Process.*, vol. 30, pp. 2771–2783, 2021.

[3] V. Mirjalili, S. Raschka, and A. Ross, "PrivacyNet: Semi-adversarial networks for multi-attribute face privacy," *IEEE Trans. Image Process.*, vol. 29, pp. 9400–9412, 2020.

[4] H. Deng, C. Han, H. Cai, G. Han, and S. He, "Spatially-invariant style-codes controlled makeup transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6545–6553.

[5] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[6] Y. Jo and J. Park, "SC-FEGAN: Face editing generative adversarial network with user's sketch and color," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1745–1753.

[7] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.

[8] M. Liu et al., "STGAN: A unified selective transfer network for arbitrary image attribute editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3668–3677.

[9] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9240–9249.

[10] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1–9.

[11] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5444–5453.

[12] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "SfSNet: Learning shape, reflectance and illuminance of faces 'in the wild,'" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6296–6305.

[13] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "GAN inversion: A survey," 2021, *arXiv:2101.05278*.