Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(7): 4889–4912.
DOI: 10.3934/era.2026216
Received: 13 December 2025
Revised: 26 April 2026
Accepted: 20 May 2026
Published: 08 June 2026

Research article

Edit-discrepancy-guided feature transformation in encoder-based GAN inversion for real image attribute editing

Wenbo Yan[1], Xing Xu[1,2,*], Yinglong Zhang[1], Xuewen Xia[1] and Yuanxiang Li[3,4]

[1] School of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China

[2] Center for China-ASEAN Regional Collaborative Development, Minnan Normal University, Zhangzhou 363000, China

[3] School of Computer Science, Wuhan University, Wuhan 430072, China

[4] Digital Strategy Development Research Institute of Hechi University, Hechi 546399, China

* **Correspondence:** Email: xx1889@mnnu.edu.cn.

Abstract: Image attribute editing based on generative adversarial networks (GANs) typically begins by mapping real images to the latent space of a pretrained StyleGAN, followed by manipulating the corresponding latent codes. However, low-rate latent codes suffer from an information bottleneck, making it challenging to faithfully reconstruct complex real images. Recent encoder-based methods enhance reconstruction by injecting high-rate features into intermediate generator layers to better preserve fine details, but they often yield misaligned details in the edited images. The primary reason is that these methods still rely on global linear transformations of high-rate features, which overlook the nonlinear and spatially localized nature of real edits. To this end, we build on a high-fidelity encoder-based GAN inversion backbone and introduce an additional adaptive feature editor that is specifically trained to convert high-rate features during editing so that fine details are correctly aligned with the edited image. The backbone refines the feature through a cross-attention mechanism and residual enhancement. Building on this, the feature editor employs a window-based cross-attention mechanism to extract a discrepancy signal between the original and edited generator features, which specifies both where to modify and what content to change. This signal is then fused into the feature through spatially adaptive modulation techniques, enabling region-selective attribute changes while preserving irrelevant details. Experiments on face and car benchmarks demonstrate that our method improves both reconstruction fidelity and editing quality compared to existing GAN inversion methods.

Keywords: image attribute editing; GAN inversion; generative adversarial networks; cross-attention; spatially-adaptive modulation

1. Introduction

Over the past few years, GANs [1] have advanced to the extent of generating visually realistic images, which has resulted in their widespread application in computer vision tasks. Among numerous generative models, the StyleGAN series [2–6] is distinguished by high-quality image synthesis and a semantically structured latent space that enables flexible control over image edits. Specifically, once an image has been inverted to the latent space [7], steering its latent vector along targeted semantic directions enables precise manipulation of attributes. However, extending this powerful editing capability to real images hinges on the accurate projection of the given image into the latent space. This critical step is known as GAN inversion [8].

Current GAN inversion methodologies are mainly grouped into two categories based on their implementation mechanisms: optimization-based and encoder-based. Typically, optimization-based methods employ gradient descent to independently optimize the latent encoding of each input image, aiming to minimize the reconstruction error between the generated image and the original image. Even though such methods usually ensure high reconstruction quality in terms of pixel fidelity, their need for time-consuming iterative optimization severely limits their practicality in real-world applications. Alternatively, encoder-based methods achieve efficient inference by training a feedforward network that maps input images to their latent codes in a single forward pass. However, this efficiency typically comes at the cost of reduced reconstruction fidelity due to inherent information bottlenecks [9]. When the encoder is constrained to produce latent codes in StyleGAN's compressed latent spaces (W or W^+), the limited dimensionality is insufficient to capture the fine details of real images. This often results in distortion when attempting to preserve high-frequency textures. To address this issue, prevailing methods adopt a feature-injection mechanism: In addition to encoding the latent code, the encoder extracts high-dimensional feature maps that are injected into appropriate generator layers (typically at intermediate resolutions) to restore high-frequency details lost to compression during encoding.

Despite the significant improvement in fidelity afforded by feature injection, the images reconstructed by these methods still exhibit observable discrepancies in fine-grained details compared with the original. More critically, during subsequent semantic editing, the injected features must vary in coordination with the latent code. Otherwise, the supplementary details will mismatch the new semantic state, leading to artifacts or even editing failures. To this end, the feature-style encoder [10] introduces a linear co-editing paradigm for latent maps: $\tilde{F} = F + G(\tilde{w}_{1:K}) - G(w_{1:K})$. Subsequent approaches such as the contrastive learning and cross-attention encoder (CLCAE) [11] and spatial-contextual discrepancy information compensation (SDIC) [12] essentially adopt the same linear formulation. Although this linear paradigm offers a simple baseline for feature co-editing, it falls short when confronted with more complex editing scenarios. This is due to three fundamental limitations: (1) Insufficient nonlinear modeling capacity: simple linear operations cannot capture the highly nonlinear transformations required for attribute editing. (2) Feature space mismatch: Linear mixing of encoder features (from real-image distribution) with generator features (from GAN prior) may produce statistical inconsistencies, leading to image artifacts. (3) Lack of spatial adaptivity: This paradigm employs global feature adjustments, lacking the necessary spatial selectivity to modify only the target region while preserving irrelevant details.

To address the aforementioned challenges of fine-grained reconstruction and nonlinear

collaborative editing, we propose a unified inversion framework for high fidelity and editability. The core idea is to decouple the acquisition of reconstruction details from their consistent transformation during editing: we first train a high-fidelity encoder E, then freeze its parameters and subsequently train an adaptive feature editor T. Specifically, E comprises a basic encoder and a feature tensor enhancement module (FTEM). The base encoder maps the input image to a pair consisting of a latent code and a feature tensor. FTEM then enhances the feature tensor via style-injected cross-attention and cascaded residual refinement, improving the reconstruction of global consistency and local details. To ensure detail consistency during editing, T follows a two-step "discrepancy extraction and fusion" design. First, the edit-discrepancy extraction module (EDEM) operates on pairs of unedited and edited generator features, using cross-attention with local windows to distill edit-discrepancy information that encodes semantic content and edited regions. This discrepancy guides how the enhanced feature should be correctly transformed. Then, the edit-discrepancy fusion module (EDFM) employs spatially adaptive normalization [13] to selectively fuse the discrepancy into the enhanced feature, producing its edited counterpart. EDFM ensures that semantic manipulation is applied only to target regions while maximally preserving details in irrelevant areas, and it effectively mitigates conflicts arising from feature domain misalignment.

Our contributions can be summarized as follows:

- We propose an encoder–editor GAN inversion framework that decouples the training objectives of high-fidelity reconstruction and editing, achieving faithful reconstructions and high-quality semantic editing.
- We design a feature-enhanced encoder that effectively enriches image details, along with an edit-discrepancy-guided adaptive feature editor that facilitates effective and precise attribute manipulation.
- Extensive experiments demonstrate the effectiveness of the proposed method, which significantly outperforms prior state-of-the-art approaches in both real image reconstruction and editing.

2. Related works

2.1. GAN inversion

GAN inversion aims to map a real image onto the generator's natural image manifold, enabling accurate reconstruction and controllable attribute editing by navigating semantic directions in the latent space. Early approaches primarily employed optimization-based inversion [14–17], which optimizes the latent code in spaces such as W or W^+ through back-propagation to minimize the reconstruction error between the input and generated images. Although such methods can achieve high reconstruction fidelity, they suffer from two significant drawbacks. First, their image-by-image optimization approach incurs high computational costs and lengthy processing times, limiting their practical application. Second, the latent code optimized solely for reconstruction accuracy often falls into overfitting, resulting in a loss of editing flexibility.

To overcome these drawbacks, research has shifted toward encoder-based inversion, which employs end-to-end feedforward networks to directly encode images into latent codes, enabling fast inference while maintaining editability.

Early encoder-based methods primarily performed inversion within the W or W^+ latent spaces. The pSp method [18] learns image-to-style mappings for direct reconstruction and image-to-image

translation, while e4e [19] constrains the predicted latent codes to remain closer to the native StyleGAN latent distribution for improved editability. ReStyle [20] further refines latent codes through an iterative residual prediction scheme, and Style Transformer [21] introduces a transformer-based encoder for image inversion and editing. At their core, these methods mainly represent real images using compact latent codes in W/W^+. Although these codes offer good editability, they suffer from inherent information bottlenecks that make it difficult to fully preserve high-frequency details. This results in reconstruction accuracy that generally falls short of optimization-based methods.

For improved fidelity, recent studies retained the W/W^+ latent code while in parallel training an auxiliary branch to extract high-rate features that recover lost details. For improved fidelity, recent studies retain the W/W^+ latent code while training an auxiliary branch to extract high-rate features that recover lost details. StyleRes [22] models and transforms residual information to improve real image editing with StyleGAN. SFE [23] introduces style-feature editing to achieve detail-rich inversion and high-quality image editing, and WarpRes [24] further aligns residual information through spatial warping for improved editing consistency. These methods show that high-rate feature representations can substantially enhance reconstruction fidelity. However, they also reveal a critical challenge: For effective semantic editing, the injected features must undergo synchronized transformation with the latent code. Otherwise, the added details may fail to align with the semantic changes, leading to artifacts or even complete editing failure. Motivated by this observation, within the framework proposed in this paper, we designed a consistency transformation mechanism for high-rate features tailored to the editing process. This ensures controllability and flexibility in attribute editing while preserving high-fidelity reconstruction.

2.2. Image editing

GAN-based image editing is commonly studied in the context of image-to-image translation [25–29], which learns attribute-conditioned mappings in pixel space via task-specific training. Although effective, such pipelines often need to be retrained when domains change. An alternative is to leverage the semantic structure already present in pretrained StyleGAN models and perform edits by manipulating their latent representations. The evolution of the StyleGAN models has witnessed progressively more interpretable semantic dimensions in their latent spaces (e.g., W, W^+, S [30]). Continuous traversal along specific directions in these spaces enables precise control over image attributes, effectively transforming an unconditional generator into a versatile image editing engine. Therefore, investigating and comprehending semantic pathways in the latent space of StyleGAN has become a research hotspot.

Supervised approaches employ attribute annotations or pretrained classifiers to directly ascertain semantic editing directions in the latent space. InterFaceGAN [31] uses separating hyperplanes to derive attribute directions. StyleFlow [32] learns attribute-conditioned latent transformations with conditional normalizing flows. Due to the high cost and time required to obtain large volumes of labeled data in real-world situations, some research has focused on exploring unsupervised methods. GANSpace [33] discovers interpretable editing directions by applying principal component analysis to GAN feature spaces, and StyleSpace [30] analyzes channel-level style parameters in StyleGAN to enable more disentangled and localized edits. These methods allow exploring open semantics beyond existing labels but typically require manual screening of candidate directions. Moreover, the rise of

multimodal technology has facilitated the evolution of image semantic editing towards more intuitive, text-driven approaches. A notable example is StyleCLIP [34] to bridge natural language and the latent space of StyleGAN, thereby enabling flexible text-guided editing without explicit attribute annotations.

Beyond GAN-based editing paradigms, recent image editing research has also evolved toward diffusion-based text-guided editing. Custom-Edit [35] first customizes a diffusion model to a target subject and then performs text-guided editing on a source image, thereby modifying target attributes while preserving subject-specific features. Energy-guided optimization [36] further formulates personalized image editing as a latent space optimization process guided by pretrained text-to-image diffusion models.

3. Method

3.1. Overview

The overall design architecture of the method presented in this paper is illustrated in Figure 1, which depicts the key processing steps from input to output.

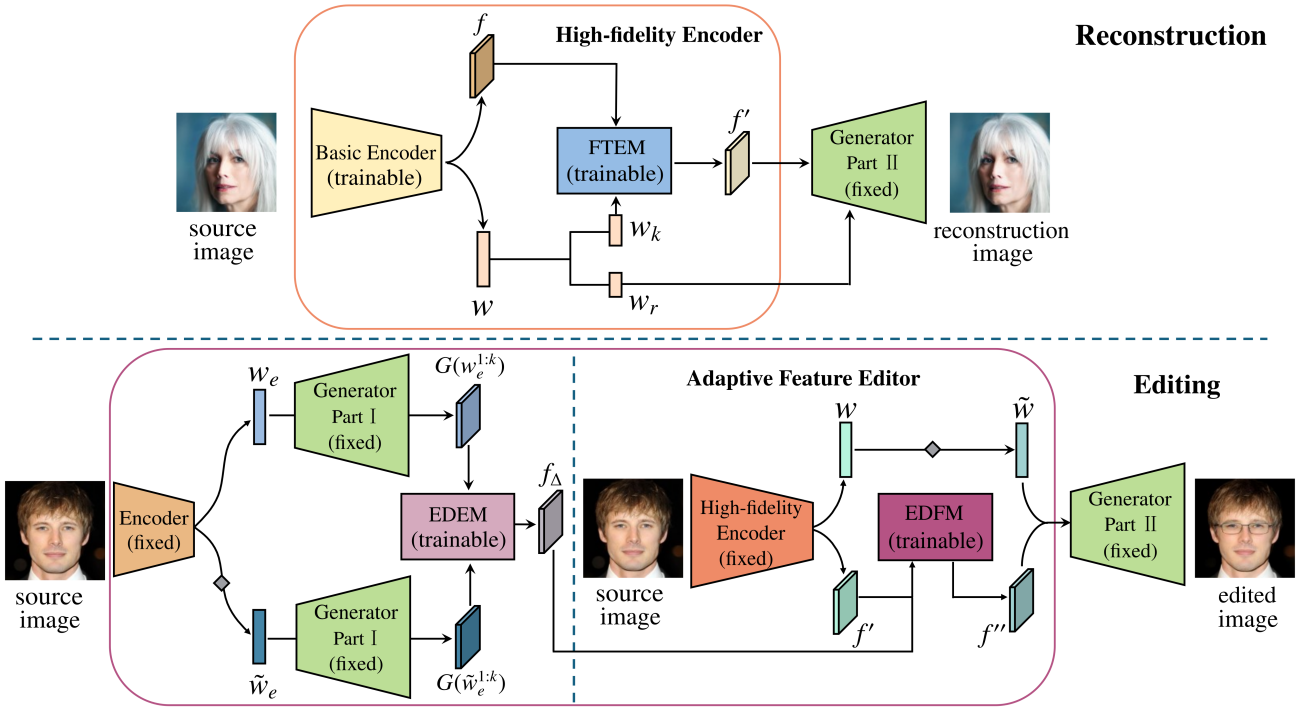

Figure 1. Our framework for high-fidelity image reconstruction and high-quality editing. The diamond symbol denotes a latent code editing operator that transforms the input code into an edited code along a specified semantic direction. The split in the StyleGAN generator is shown purely for visualization purposes.

For the source image X, the base encoder E_0, which is a feature-style encoder [10] capable of outputting higher-resolution feature tensors, first predicts the latent code w in the $W^+ \in \mathbb{R}^{N\times 512}$ space

and the feature tensor f in the StyleGAN feature $F_k \in \mathbb{R}^{C\times H\times W}$ space. The latent code w is split into two components: The first k dimensions $w_k = \{w_i \mid i = 1, 2, \cdots, k,\ w_i \in \mathbb{R}^{512}\}$ serve as input to the feature tensor enhancement module, and the remaining part $w_r = \{w_i \mid i = k+1, k+2, \cdots, N,\ w_i \in \mathbb{R}^{512}\}$ is directly fed into the generator. The values of N, C, H, W, and k are determined by the output image resolution (e.g., $N = 18$, $C = 512$, $H = 64$, $W = 64$, $k = 10$ when the output image size is 1024×1024). Subsequently, the FTEM enhances the initial feature tensor f, improving its overall structural and local detail representation capabilities, and outputs the enhanced feature tensor f'. By feeding f' together with w_r into the pretrained StyleGAN generator, high-fidelity reconstructed images are synthesized. Rich information from the source image resides in f', making it instrumental for recovering fine image details. However, if these details are not transformed accordingly during the editing process, they will not be positioned correctly on the edited image.

To this end, we introduce an adaptive feature editor T that is able to spatially adapt f' based on semantic editing. T comprises two modules: the EDEM and the EDFM. The EDEM provides an edit-discrepancy signal f_Δ containing information about the edited content and its active region, which guides the transformation of f'. The EDFM adaptively transforms f' based on f_Δ to obtain its edited version f''. Finally, f'' is fed into the generator alongside $\tilde{w}_r$ (the edited version of w_r) to synthesize high-quality edited images.

3.2. Architecture

3.2.1. High-fidelity encoder

Basic encoder. We follow the feature-style encoder [10] design for the basic encoder E_0: IResNet-50 serves as the feature extraction backbone, with two prediction branches attached to respectively output a latent code and a feature tensor. The architecture of the basic encoder is shown in Figure 2.

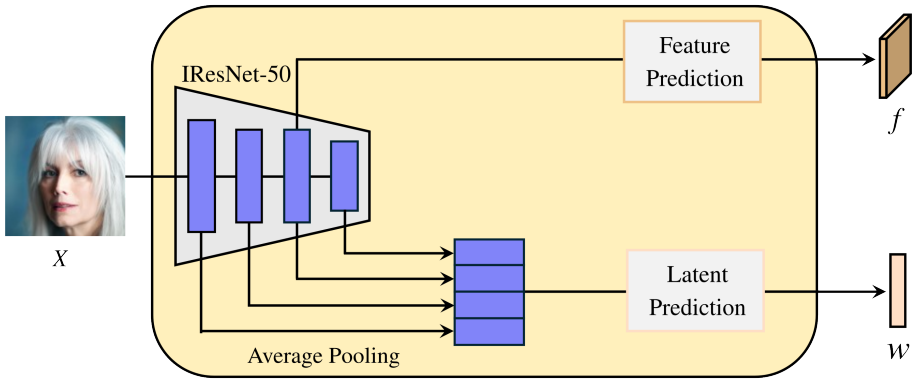

Figure 2. Basic encoder architecture.

The backbone network first extracts four intermediate features from the input image X, then pools them to the same dimension and concatenates them. These concatenated features are mapped into a latent code $w \in W^+$ through a prediction branch composed of linear layers. The third intermediate feature is also passed to a feature tensor output branch constructed by a convolutional neural network to encode the feature tensor $f \in F_k$. In the original feature-style encoder, when synthesizing 1024×1024 face images, the predicted feature tensor has size $512\times16\times16$, mainly capturing coarse-scale semantics

(e.g., facial contours).

To better represent details such as skin texture, we increase the spatial resolution of the feature tensor from 16×16 to 64×64. To this end, the IResNet-50 architecture is minimally adjusted by changing the stride of the first convolution and the downsampling operation in the third residual stage from 2 to 1, while keeping the remaining configuration identical to the original version. Given an input image X, the base encoding process can be expressed as follows:

$$(w,\ f) = E_0(X), \tag{3.1}$$

where $w \in \mathbb{R}^{18\times512}$, $f \in \mathbb{R}^{512\times64\times64}$ denote the latent code and the feature tensor predicted by E_0, respectively.

Feature tensor enhancement module. To further improve the image representations generated by the model, we design a feature tensor enhancement module composed of a cross-attention fusion block and a residual refinement block, as illustrated in Figure 3.

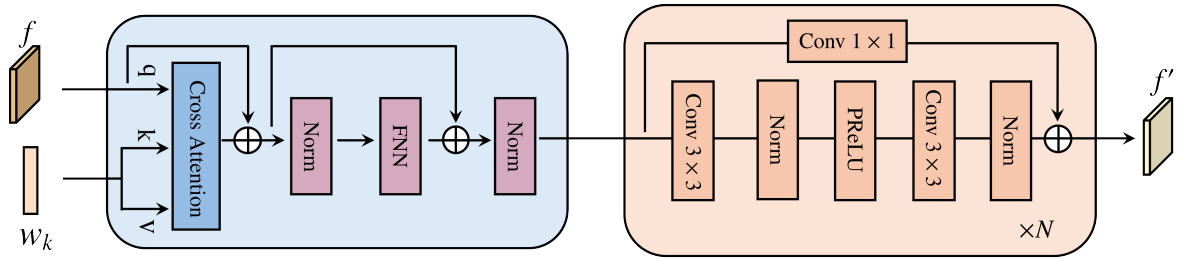

Figure 3. Network architecture of the feature tensor enhancement module.

We first perform cross-attention between the first k dimensions latent code w_k and the feature tensor f, outputting a residual to enhance f. In the cross-attention block, f serves as the query Q, and w_k provides both the key K and value V. This setup allows f to assimilate the global structural information from w_k, ultimately improving the perceptual quality of the reconstruction. The output of the cross-attention is then refined by a lightweight residual network to progressively enhance local details, yielding the enhanced feature tensor f'. Formally, the expression for the overall enhancement process is:

$$\begin{gathered} Q = fW_Q, \quad K = w_kW_K, \quad V = w_kW_V, \\ \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \\ f' = \text{Res}(\text{Attention}(Q, K, V) + f), \end{gathered} \tag{3.2}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{512\times512}$ are learnable projection matrices, and d denotes the feature dimension. In practice, we employ a multihead attention mechanism for parallel computation. Res$(\cdot)$ represents a residual network stacked from multiple blocks. Each block contains two 3×3 convolutional layers with a PReLU activation in between.

Thus, the encoding process of the high-fidelity encoder E can be summarized as follows:

$$(w, f') = E(X). \tag{3.3}$$

Finally, the reconstructed image X_{rec} is generated by the pretrained StyleGAN generator G, which takes the enhanced feature tensor f' and the latter latent code portion w_r as its input:

$$X_{rec} = G(f', w_r). \tag{3.4}$$

3.2.2. Adaptive feature editor

Edit-discrepancy extraction module. This module aims to provide accurate transformation guidance (i.e., edit-discrepancy) for the enhanced feature f' generated during the reconstruction stage. Because f' replaces the output after the kth convolutional layer in the StyleGAN generator, to maintain feature space consistency, the extraction of edit-discrepancy should be performed in the StyleGAN feature space F_k. To achieve this objective, we first employ a pretrained encoder (e4e is selected in this work) that demonstrates excellent editing performance in the W^+ space to invert real images into latent codes w_e. Then, we apply an editing strength α along a specific attribute direction Δw to obtain the edited latent code $\tilde{w}_e = w_e + \alpha \cdot \Delta w$. By feeding both the original latent code w_e and the edited latent code $\tilde{w}_e$ into the generator G, we extract their corresponding features at layer k: $G(w_e^{1:k})$ and $G(\tilde{w}_e^{1:k})$.

Because attribute editing usually influences only localized regions of an image, the module is designed based on a Swin transformer [37] architecture equipped with cross-attention, as shown in Figure 4. It accurately delineates both the altered content and the corresponding impacted regions within the feature space, based on the difference between original and edited features.

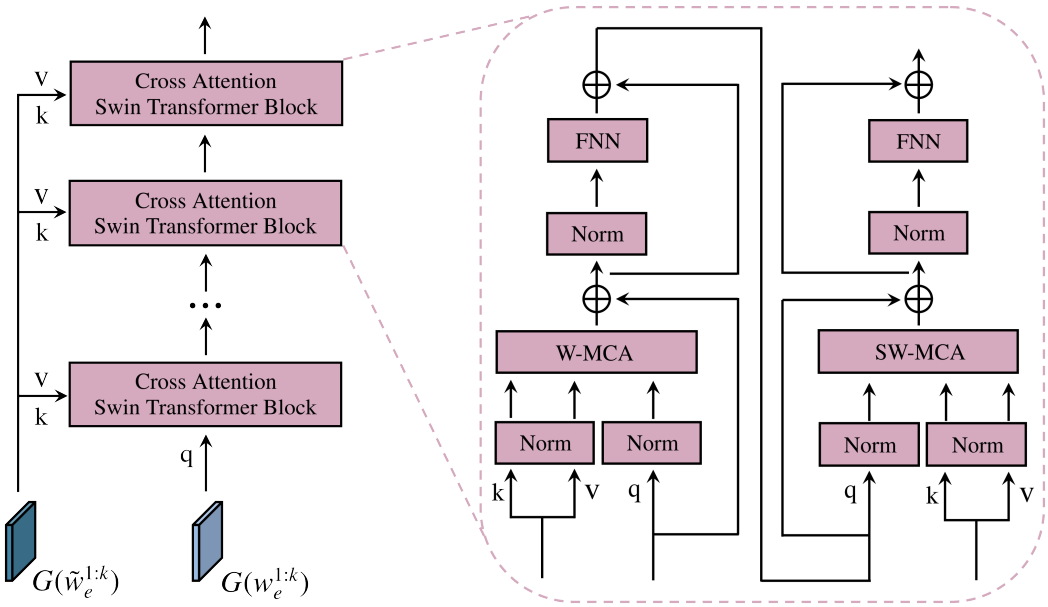

Figure 4. Network architecture of the EDEM. W-MCA, and SW-MCA are multihead cross-attention modules with regular and shifted windowing configurations, respectively.

In details, given input features $G(w_e^{1:k}) \in \mathbb{R}^{C\times H\times W}$ and $G(\tilde{w}_e^{1:k}) \in \mathbb{R}^{C\times H\times W}$, the cross-attention Swin transformer layer first partitions them into nonoverlapping local windows of size $M \times M$, resulting in HW/M^2 windows in total. Then, for each pair of co-located windows, local cross-attention is computed by taking the queries Q from $G(w_e^{1:k})$ and the keys K and values V from $G(\tilde{w}_e^{1:k})$. The single-head attention is computed as follows:

$$Q = G(w_e^{1:k})P_Q, \quad K = G(\tilde{w}_e^{1:k})P_K, \quad V = G(\tilde{w}_e^{1:k})P_V,$$
$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V, \tag{3.5}$$

where P_Q, P_K, $P_V \in \mathbb{R}^{C\times d}$ are shared linear projection matrices, $d = C/h$ denotes the feature dimension perhead, h is the number of attention heads, and B represents the relative position bias learned separately for each head. In practice, we employ a multihead attention mechanism for parallel computation.

Taking into account that the scope of the attribute editing may span multiple adjacent windows, we alternate between regular and shifted window partitioning. The shifted window partitioning means features are shifted by ($\lfloor M/2 \rfloor$, $\lfloor M/2 \rfloor$) pixels [38] along height and width. They are then partitioned, attended, and shifted back. Each attention layer employs Pre-LN and a residual path, succeeded by a two-layer multilayer perceptron (MLP) that uses Gaussian error linear unit (GELU) as its intermediate activation.

After merging all the windows, we obtain an output feature with the same shape as the input feature. Then, subtract the original input $G(w_e^{1:k})$ from this output to obtain the residual representing editing content changes:

$$d = E_{cs}(G(w_e^{1:k}), G(\tilde{w}_e^{1:k})) - G(w_e^{1:k}), \tag{3.6}$$

where $E_{cs}(\cdot, \cdot)$ denotes a stack of cross-attention Swin transformer layers.

Meanwhile, we derive a spatial gating g_m from the attention tensor *Attn* at the final layer to highlight where editing should be applied: Take a headwise mean, and then for each query, take the max over keys, map back to $H\times W$, and pass through a sigmoid activation. The final edit-discrepancy is computed as the Hadamard product between the residual and the spatial gating:

$$f_\Delta = d \odot g_m. \tag{3.7}$$

Edit-discrepancy fusion module. After obtaining the edit-discrepancy f_Δ, it needs to be accurately fused into the enhanced feature f'. Spatially adaptive normalization (SPADE) [13] has shown that simply summing or concatenating the semantic layouts with the original features and then applying a Conv-Norm-Activation stack tends to "wash away" the semantic information due to the effect of the normalization layer. Inspired by SPADE, we treat f_Δ as a binary spatial layout (edited vs. nonedited) and use spatially adaptive normalization to explicitly modulate intermediate activations, allowing edit semantics to be delivered effectively across the network.

Based on this insight, we design a feature modulation network built upon SPADE, which consists of multiple residual blocks employing spatially adaptive normalization, as illustrated in Figure 5. Specifically, the ith residual block takes f_Δ as a shared conditional feature and predicts per-pixel modulation parameters γ_i, β_i, and gating parameters δ_i:

$$\gamma_i = H_i^\gamma(f_\Delta), \quad \beta_i = H_i^\beta(f_\Delta), \quad \delta_i = H_i^\delta(f_\Delta), \tag{3.8}$$

where H_i^γ, H_i^β, H_i^δ are each implemented as two 3×3 convolutional layers, separated by a GELU activation. Subsequently, the input feature of this block, denoted by $f_i \in \mathbb{R}^{C\times H\times W}$ with $f_1 = f'$,

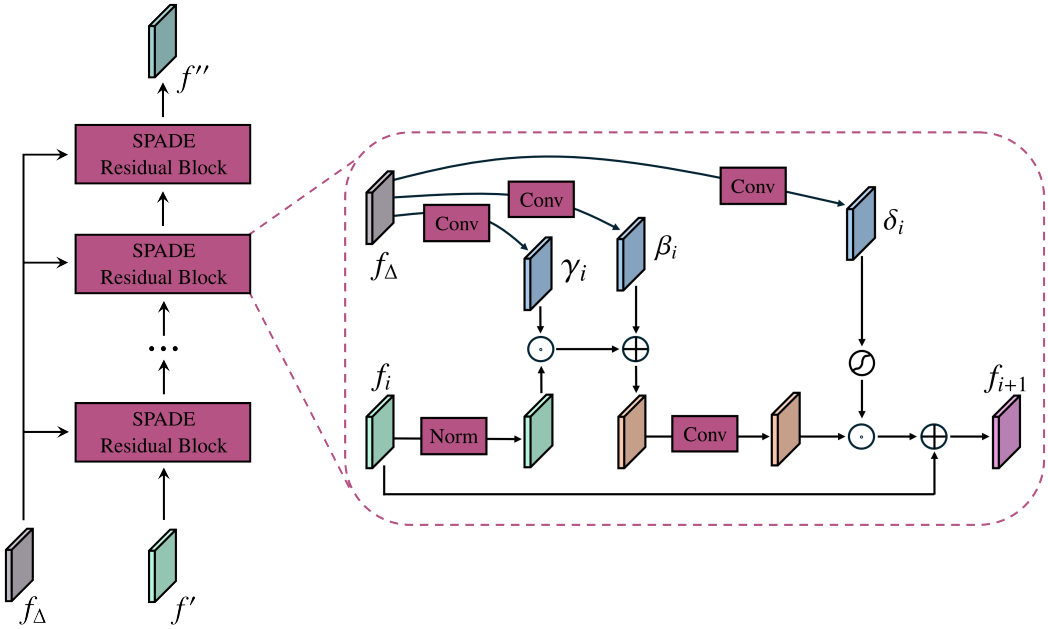

Figure 5. Network architecture of the EDFM.

is modulated by γ_i and β_i, after which a spatial gate $\sigma(\delta_i)$ adaptively determines where the residual should be injected to produce the feature of the next layer:

$$f_{i+1} = f_i + \sigma(\delta_i) \odot H_i(\text{Norm}(f_i) \odot (1 + \tanh(\gamma_i)) + \beta_i), \tag{3.9}$$

where $\sigma(\cdot)$ denotes the sigmoid function, and $H_i(\cdot)$ is implemented as two 3×3 convolutional layers interleaved with a GELU activation. Norm$(\cdot)$ denotes normalization without affine parameters. The factor $(1 + \tanh(\cdot))$ keeps the modulation near an identity mapping at initialization, improving training stability. After several rounds of modulation, we obtain the edited version of f', denoted as f''.

The complete transformation of the adaptive feature editor T is formulated as:

$$f'' = T(G(w_e^{1:k}), G(\tilde{w}_e^{1:k}), f'). \tag{3.10}$$

The edited image X_{edit} is generated by the pretrained StyleGAN generator G, which takes the edited feature tensor f'' and the edited latent code $\tilde{w}_r = w_r + \alpha \cdot \Delta w$ as its input:

$$X_{\text{edit}} = G(f'', \tilde{w}_r). \tag{3.11}$$

3.3. Training objectives

Training of encoder. The training objective of the encoder E is to accurately invert an input image X so as to obtain a reconstructed image $X_{rec} = G(f', w_r)$. To ensure that the latent code w effectively supports existing latent space editing, we generate another reconstruction $X_{rec\text{-}w} = G(w)$ obtained from w only. During training, we compute several commonly used losses between the reconstructed images and the ground-truth image, including the pixel loss L_2, the perceptual similarity loss L_{lpips}, and the identity-preserving loss L_{id}. They are defined as:

$$L_2 = \|X_{rec} - X\|_2 + \|X_{rec\text{-}w} - X\|_2, \tag{3.12}$$

$$L_{lpips} = \|V(X_{rec}) - V(X)\|_2 + \|V(X_{rec\text{-}w}) - V(X)\|_2, \tag{3.13}$$

where $V(\cdot)$ refers to a pretrained visual geometry group (VGG) network employed for feature extraction.

$$L_{id} = (1 - \langle F(X_{rec}), F(X)\rangle) + (1 - \langle F(X_{rec\text{-}w}), F(X)\rangle), \tag{3.14}$$

where $F(\cdot)$ is instantiated as the pretrained ArcFace network on facial data, and as the ResNet-50 network trained with MoCov2 for other domains. $\langle\cdot,\cdot\rangle$ is the cosine similarity. To encourage the reconstructed images to look more realistic, we employ an adversarial loss:

$$L_{adv} = -\mathbb{E}[log(D(X_{rec}))] - \mathbb{E}[log(D(X_{rec\text{-}w}))], \tag{3.15}$$

where D is the StyleGAN discriminator, whose parameters are first obtained from a pretrained StyleGAN model and are further updated during our training. At the same time, we employ a feature regularization loss to constrain the enhanced feature f', preventing it from deviating excessively from the StyleGAN feature space:

$$L_{f_{rec}} = \|f' - G(w_k)\|_2. \tag{3.16}$$

The overall reconstruction loss is given as the weighted sum of the above terms:

$$L_{rec} = \lambda_{l2}L_2 + \lambda_{lpips}L_{lpips} + \lambda_{id}L_{id} + \lambda_{adv}L_{adv} + \lambda_{f_{rec}}L_{f_{rec}}. \tag{3.17}$$

Training of feature editor. The training objective of the feature editor T is to transform the encoder-predicted feature f' into its edited counterpart f'' under a given latent manipulation, so that the synthesized output matches the target semantic edit. To construct supervision, we first invert a ground-truth image X with e4e [19] to obtain w_e and its reconstruction $X^e_{rec} = G(w_e)$. We then move along a semantic direction Δw with strength α to get the edited latent code $\tilde{w}_e = w_e + \alpha \cdot \Delta w$ and the corresponding target edited image $X^e_{edit} = G(\tilde{w})$. Next, we invert X_{rec} with E to obtain its reconstruction code w and feature f'. Following the same manipulation on the latent path, we take the latter part edited latent code and denote it by $\tilde{w}_r$. T converts f' into f'', and the output image is synthesized as $X_{edit} = G(f'', \tilde{w}_r)$. We compute the pixel L_2, perceptual L_{lpips}, and identity L_{id} losses between X^e_{edit} and X_{edit}. Notably, we keep the parameters of encoder E fixed during the training of feature editor, as E is considered to be already adequately trained. In addition, relying only on synthetic edited targets may degrade reconstruction fidelity on real images. Therefore, during training, we also treat X as a target by setting the editing strength $\alpha = 0$ (i.e., no edit). The corresponding losses are

$$L_2 = \|X_{edit} - X^e_{edit}\|_2 + \|X_{edit} - X\|_2, \tag{3.18}$$

$$L_{lpips} = \|V(X_{edit}) - V(X^e_{edit})\|_2 + \|V(X_{edit}) - V(X)\|_2, \tag{3.19}$$

$$L_{id} = (1 - \langle F(X_{edit}), F(X^e_{edit})\rangle) + (1 - \langle F(X_{edit}), F(X)\rangle). \tag{3.20}$$

In addition to the aforementioned losses, we propose an edit-discrepancy regularized loss. It aligns the learned feature change $(f'' - f')$ with the discrepancy f_Δ predicted by the EDEM, encouraging the editor to produce the desired local modification while preserving nonedited regions:

$$L_{f_{edit}} = \|f_\Delta - (f'' - f')\|_2. \tag{3.21}$$

The overall editing loss is given as the weighted sum of the above terms:

$$L_{edit} = \lambda_{l2}L_2 + \lambda_{lpips}L_{lpips} + \lambda_{id}L_{id} + \lambda_{f_{edit}}L_{f_{edit}}. \tag{3.22}$$

4. Experiments

4.1. Experiment settings

Datasets. To evaluate our method, we employ datasets from two distinct visual domains: human faces and cars. In the facial domain, model training uses the Flickr-Face HQ (FFHQ) dataset [3], and evaluation is performed on CelebA-HQ [2], consistent with prior work. For the car domain, the Stanford Cars dataset [39] is used with its standard split for training and testing.

Baselines. We compare our approach against encoder-based inversion methods, including e4e [19], high-fidelity GAN inversion (HFGI) [40], feature-style encoder (FS) [10], CLCAE [11], StyleRes [22], SDIC [12], and style-feature editor (SFE) [23]. For faces, all methods provide official models, and we report results using their released checkpoints. For cars, because several models are unavailable, we evaluate the available e4e, CLCAE, and SDIC. To enhance the completeness of the comparison, we additionally included HyperStyle [41] for evaluation.

Implementation details. In our experiments, the pretrained generator used was derived from StyleGAN2 [4]. For faces, images are synthesized at 1024×1024; for cars, at 512×384. During the training of the editor, we use InterFaceGAN [31] (pose, smile, age, makeup, and glasses), StyleSpace [30] (gender and blond hair), and StyleCLIP [34] (angry, afro, bobcut hairstyle, Mohawk hairstyle, and purple hair) to edit facial attributes; GANSpace (grass, color, and cube shape) and StyleSpace (trees and headlights) to edit car attributes. For the reconstruction stage, we set λ_{l2}, λ_{lpips}, λ_{id}, λ_{adv}, and $\lambda_{f_{rec}}$ to 1.0, 0.8, 0.1, 0.01, and 0.01, respectively. We enable the adversarial loss only after 45,000 iterations. The total number of training iterations for this stage is 120,000. For the editing stage, we set λ_{l2}, λ_{lpips}, λ_{id}, and $\lambda_{f_{edit}}$ to 1.0, 0.8, 0.1, and 0.6, respectively. We enable the edit-discrepancy loss only after 20,000 iterations. The total number of iterations for this stage is 45,000. For the module configuration, FTEM adopts a residual refinement network with six residual blocks, where each block uses a constant channel width of 512. In EDEM, the cross-attention Swin transformer consists of two layers and uses a window size of 16×16. For EDFM, the SPADE-based fusion network employs six residual blocks, each with a constant channel width of 512. Both stages use the Ranger optimizer with a learning rate of 0.0001. All experiments are executed on an NVIDIA GeForce RTX 3090 GPU with a batch size of four.

4.2. Reconstruction results

Qualitative evaluation. Figure 6 presents a qualitative comparison of reconstruction performance between our method and several state-of-the-art GAN inversion approaches. Because e4e performs inversion solely in the W^+ space, the resulting latent code carries insufficient image information, leading to severe distortion. FS and CLCAE enhance inversion via feature injection, but both operate on relatively low-resolution latent maps (e.g., $f \in \mathbb{R}^{512\times16\times16}$), whose limited dimension is insufficient for detail reconstruction. Although HFGI and SDIC increase the latent map resolution to $512 \times 64 \times 64$, both adopt e4e for initial reconstruction before refining details. Due to the inherent distortion limitations of the initial reconstruction image, the final image lacks fine-grained details (e.g., hair strands and accessories). Notably, HFGI even exhibits deviations from the original image in terms of global perception. The reconstruction results of StyleRes are better than the methods mentioned above, but still cannot achieve precise reconstruction of the source image. Although SFE can

accurately reconstruct the source image, the reconstructed image shows a mesh texture, especially on the skin and hair, resulting in the inverted image being not realistic enough. By comparison, our approach produces higher-quality reconstructions, achieving better overall visual perception while preserving finer details.

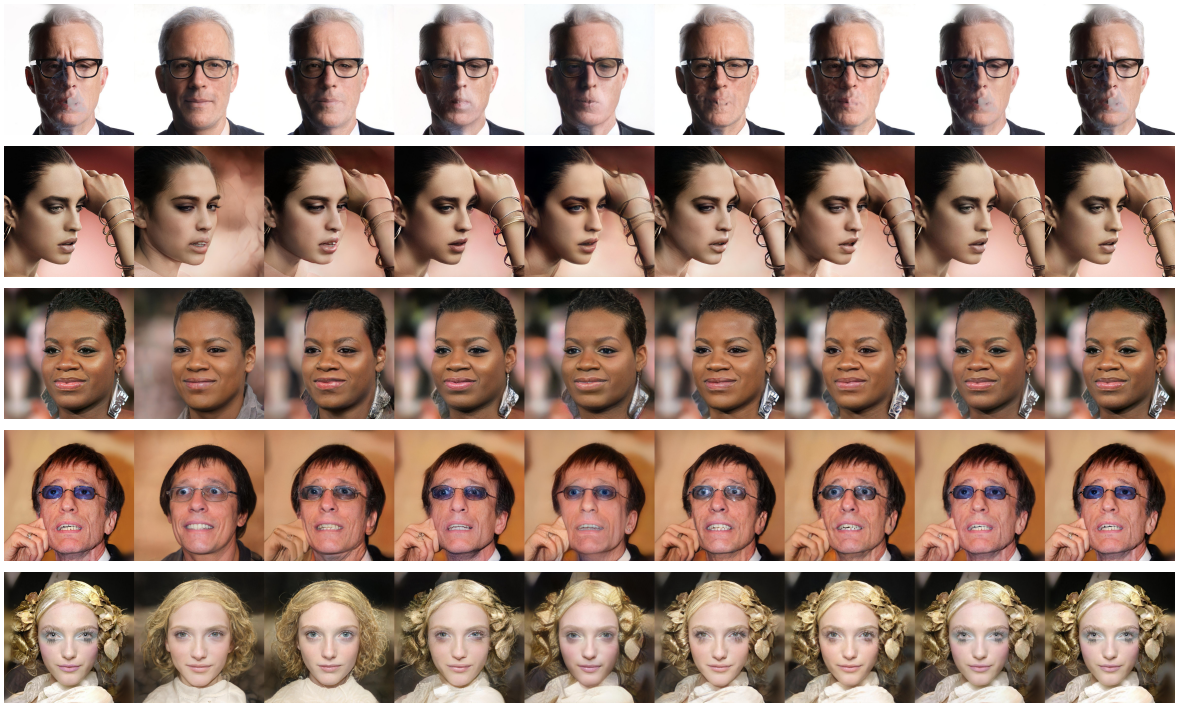

Figure 6. Qualitative reconstruction comparison of our method with baselines on CelebA-HQ dataset.

For instance, in the first row, our method faithfully reconstructs the smoke particles near the lips, whereas other methods (except SFE) either smooth out the smoke, causing it to disappear, or they force reconstructions that distort the skin texture around the mouth. For other rows, our method is observed to produce clearer facial contours (side profile without artifacts in the second row) and skin textures (the mole on the left cheek in the fourth row), along with more faithful reconstruction of accessories (the earrings in the third row) and makeup (the eye makeup color in the last row). Beyond that, in the car domain, our reconstruction quality is also superior, as shown in Figure 7.

Quantitative evaluation. We employ multiple metrics to quantify the reconstruction quality of our method and compare it with baselines. All test results are obtained on the top 2000 images from the respective datasets, as adopted by most existing methods. Specifically, we use L2, learned perceptual image patch similarity (LPIPS) [42] and multi-scale structural similarity index measure (MS-SSIM) [43] to measure pixel-level, feature-level, and structural similarity between the reconstructed image and the original image, respectively. We also extract features with CurricularFace [44] and compute their cosine similarity as the identity score (ID), which measures how well each reconstructed image preserves the identity of the original input. In addition, we assess the realism of synthesized images by computing the Fréchet inception distance (FID) [45] value between the distributions of real images and inverted images. Table 1 summarizes the quantitative

Figure 7. Qualitative comparison of our method with baselines on the Stanford Cars dataset.

Table 1. Quantitative comparison results of our method with baselines on the CelebA-HQ dataset.

	Reconstruction quality						**Editing quality - ID**				
Method	L2 (↓)	LPIPS (↓)	ID (↑)	FID (↓)	MS-SSIM (↑)	PSNR (↑)	Smile	Age	Glass	Pose	User study (↓)
e4e [19]	0.0448	0.190	0.509	29.969	0.619	19.196	0.471	0.317	0.333	0.489	93%
FS [10]	0.0114	0.063	0.805	12.535	0.748	23.657	0.544	0.584	0.582	0.706	79%
HFGI [40]	0.0199	0.111	0.686	18.907	0.714	22.085	0.596	0.422	0.414	0.604	87%
CLCAE [11]	0.0105	0.079	0.721	22.467	0.755	24.275	0.624	0.580	0.478	0.711	80%
StyleRes [22]	0.0120	0.071	0.758	10.724	0.791	23.545	0.653	0.500	0.529	0.639	71%
SDIC [12]	0.0060	0.053	0.873	13.502	0.816	25.569	0.724	0.538	0.597	0.697	64%
SFE [23]	0.0019	0.020	0.953	5.760	**0.898**	28.418	0.809	0.630	0.682	0.807	61%
Ours	**0.0016**	**0.016**	**0.956**	**4.734**	0.897	**28.488**	**0.824**	**0.636**	**0.705**	**0.838**	–

reconstruction results in the facial domain, with the best and second-best scores for each metric highlighted in bold and underlined, respectively. As shown, our method surpasses all previous approaches across every metric except MS-SSIM. In particular, compared with the second-best method, SFE (0.0019 for L2 and 0.0020 for LPIPS), we achieve relative reductions of approximately 16% and 20%, respectively, indicating more precise recovery of fine-grained details. The FID score is also greatly improved, by nearly 18% relative to the second-best, showing that our inverted images are much closer to real ones. We further evaluate our method on the Stanford Cars dataset, as shown in Table 2, and observe similarly strong results.

Table 2. Quantitative comparison results of our method with baselines on the Stanford Cars dataset.

	Reconstruction quality					**Editing quality - FID**		
Method	L2 (↓)	LPIPS (↓)	FID (↓)	MS-SSIM (↑)	PSNR (↑)	Grass	Color	User study (↓)
e4e [19]	0.1195	0.318	16.683	0.476	15.222	19.288	30.273	92%
HyperStyle [41]	0.0717	0.270	13.115	0.550	17.373	17.640	26.940	78%
CLCAE [11]	0.0305	0.156	14.956	0.680	20.548	21.058	27.773	83%
SDIC [12]	<u>0.0210</u>	<u>0.108</u>	<u>11.378</u>	<u>0.765</u>	<u>21.866</u>	<u>14.126</u>	<u>25.734</u>	66%
Ours	**0.0022**	**0.028**	**4.378**	**0.943**	**30.072**	**7.545**	**15.974**	–

4.3. Editing results

Qualitative evaluation. Figure 8 presents the visual outcomes of facial attribute editing. Editing directions are obtained from InterFaceGAN [31]. We demonstrate comparative outcomes for four attributes: smile, age, pose, and glasses. The first three attributes are manipulated in both positive (enhancement) and negative (weakening) directions; only the addition of glasses is shown.

Thanks to our designed adaptive feature editor, the model demonstrates satisfactory performance in attribute editing. Taking pose rotation as an example, our method accurately achieves target-angle rotation transformations while avoiding artifact generation. The generated results appear more visually natural, outperforming existing methods. Moreover, our model not only achieves the desired attribute editing but also preserves rich detail from the source image: The shape and skin tone on the back of the hand in the smile↑ row, the earrings in smile↓, the closed eyes in the glasses row. Additionally, we have demonstrated the edited results in the car domain, as shown in Figure 7. Editing directions are obtained from GANSpace [33]. In summary, our method enables flexible attribute editing while preserving the original details in unedited regions.

Quantitative evaluation. There is no standardized metric for precisely quantifying image editing quality. We therefore adopt the proxy evaluation methods widely used in prior work: ID between original and edited images for faces based on the premise that editing should preserve personal identity, and FID for cars. We further conduct a user study to complement these metrics with human perceptual evaluations. We randomly select 36 images and divide them into six groups, each containing four face images and two car images. For each original image, 25 invited participants are shown two edited versions (one generated by our method and the other by a baseline method) and are asked to choose the one with superior editing quality and fidelity. For each comparison method, we calculate the proportion of votes cast for our method across all participants in its corresponding group, which serves as the user

preference ratio. Tables 1 and 2 present the results of the aforementioned evaluation in the face and car domains, respectively. As shown in the tables, our proposed method outperforms previous approaches in editing quality across both domains. Concurrently, the user study indicates that the majority of participants favor our method.

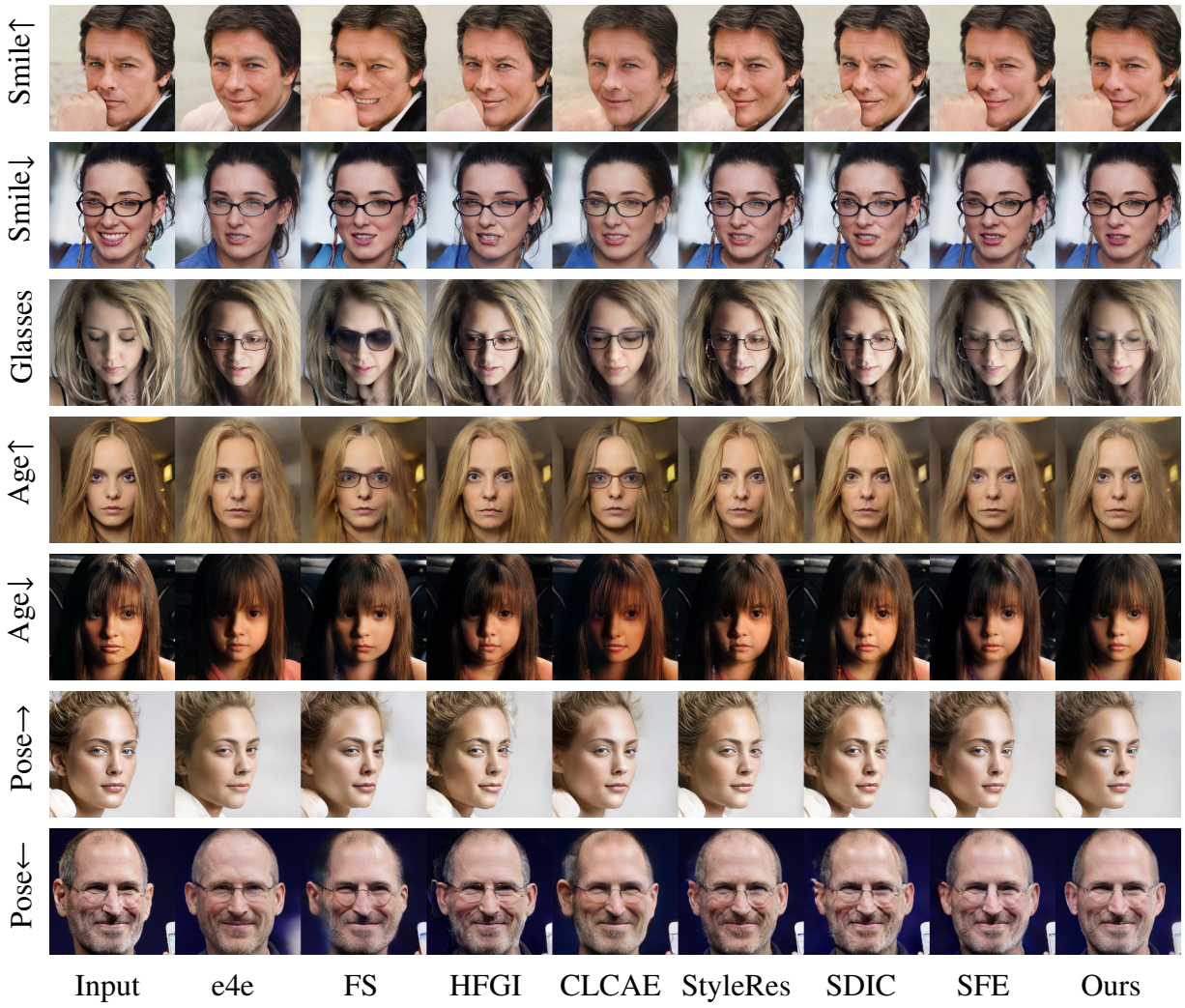

Figure 8. Qualitative editing comparison of our method with baselines on the CelebA-HQ dataset.

4.4. Ablation study

Effect of FTEM. To assess the contribution of the feature tensor enhancement module, we evaluate its necessity by comparing the reconstruction quality when the module is retained versus when it is removed. In the absence of the feature enhancement module, the latent encoder and the feature tensor predicted by the basic encoder are directly fed into the generator for image reconstruction. Visual comparison results are shown in Figure 9(a).

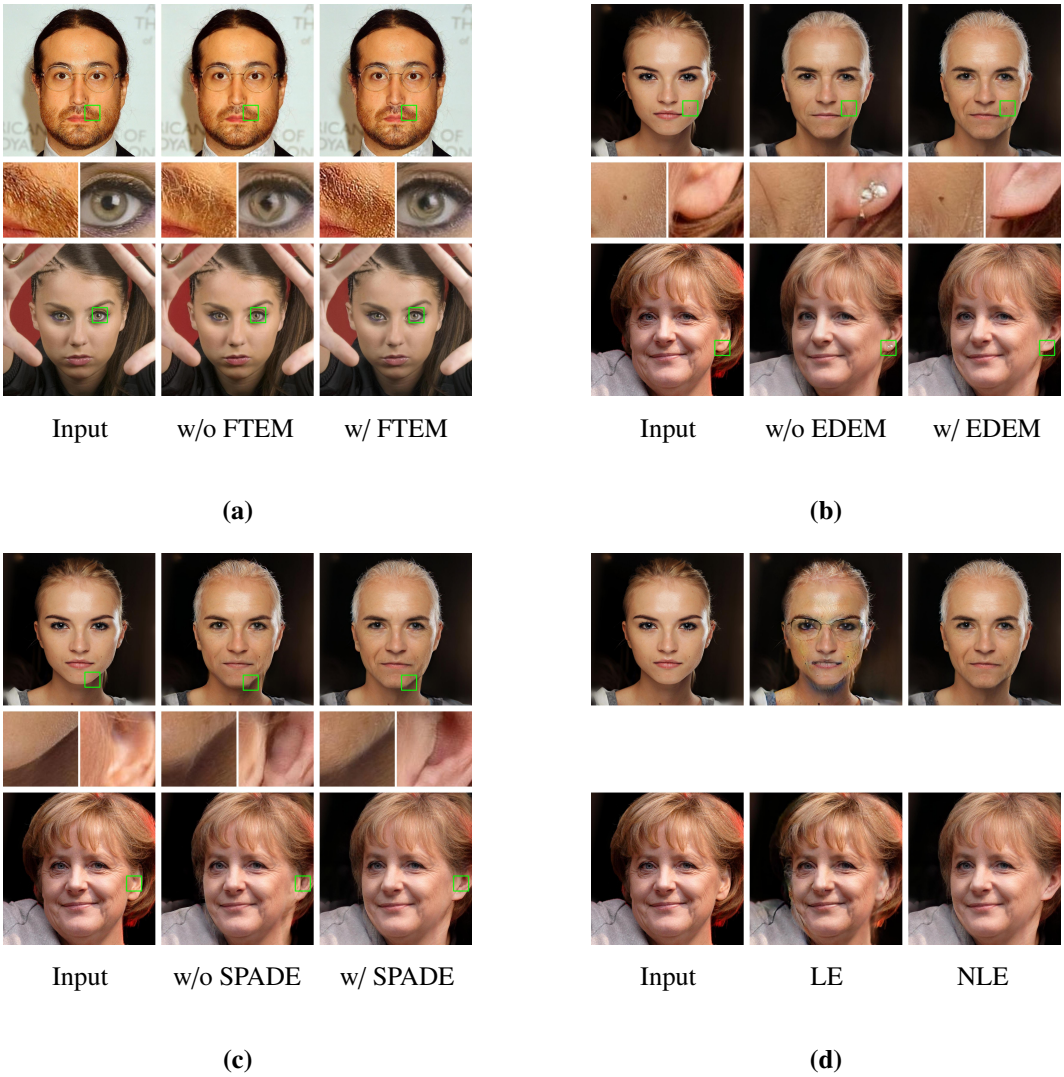

Figure 9. Visualization results from different ablation experiments on the face domain. (a) compares reconstructed images. (b), (c), and (d) compare edited images, where the top row shows aging edits, and the bottom row shows pose-rotation edits. "LE" and "NLE" denote linear and nonlinear editing of the feature tensor, respectively.

The overall image structures generated by both configurations exhibit high similarity, yet differences persist in detail reconstruction. In the version without the feature tensor enhancement module, areas such as the eye reflections and beard texture show reduced fidelity, whereas incorporating this module enables finer restoration of these high-frequency details. This demonstrates the critical role of the feature tensor enhancement module in improving fine-grained reconstruction quality. The changes in reconstruction metrics in Table 3 also corroborate this.

Effect of EDEM. To leverage edit-discrepancy information for guiding feature tensor transformations, we employ an EDEM. To validate whether edit-discrepancy truly guides feature transformation, we adopt the raw feature difference between the edited and reconstructed images at generator layer k while removing the edit-discrepancy regularization loss.

Table 3. Quantitative comparison of reconstruction results with and without the feature tensor enhancement module.

Config	L2 (↓)	LPIPS (↓)	ID (↑)	FID (↓)	MS-SSIM (↑)	PSNR (↑)
W/o FTEM	0.0026	0.021	0.949	5.134	0.886	27.668
W/ FTEM	**0.0016**	**0.016**	**0.956**	**4.734**	**0.897**	**28.488**

As shown in Figure 9(b), the absence of the edit-discrepancy extraction module's guidance leads to unintended alterations in irrelevant regions, violating the principle that edits should preserve original content as much as possible. Conversely, incorporating this module effectively prevents such issues, preserving details in irrelevant areas.

Effect of SPADE in the EDFM. The role of the edit-discrepancy fusion module is to accurately transform feature tensors, where SPADE [13] technology ensures regional selectivity in editing. To validate the necessity of this design, we compare a simplified version without SPADE achieved by directly concatenating the edit-discrepancy map with the feature tensor and feeding it into a residual network for feature transformation. As shown in Figure 9(c), the absence of SPADE induces artifacts in edited images, stemming from the failure to correctly transform the feature tensor within the corresponding editing region. In contrast, the complete model generates visually natural images while achieving the desired editing outcome. This experiment confirms the critical role of SPADE in editing precision.

Effect of the adaptive feature editor. We also compare the linear feature editing method in FS [10] with our proposed method, which edits the reconstructed feature tensor using linear additive operations instead of the adaptive feature editor. The resulting edits are shown in Figure 9(d). It can be observed that linear editing fails to effectively transform features in high-dimensional space, resulting in significant ghosting artifacts in the edited images. Instead, our method employs adaptive nonlinear feature transformation, enabling not only high-quality attribute editing but also preserving greater detail.

Similar to the comparison experiments, we not only evaluate each module's role through subjective visual perception but also quantitatively assess its contribution from the perspective of identity consistency (ID) across multiple attribute editing tasks. Detailed comparison results are presented in Table 4. It can be observed that in editing tasks, the complete model consistently achieves higher identity similarity scores than all ablation configurations, demonstrating particularly significant improvements compared to the linear editing configuration. These qualitative and quantitative results fully validate the positive contributions made by each design module in enhancing both model reconstruction quality and editing performance.

Table 4. Quantitative editing comparison of different ablations on the CelebA-HQ dataset.

Config	Smile	Age	Glasses	Pose
W/o EDEM	0.810	0.603	0.645	0.798
W/o SPADE	0.817	0.611	0.652	0.762
LE	0.772	0.562	0.598	0.729
Final model	**0.824**	**0.636**	**0.705**	**0.838**

4.5. Limitations

One limitation of the proposed method lies in its computational overhead. As shown in Table 5, we report the total number of parameters required for inference (including those of the StyleGAN2 generator), the giga multiply-accumulate operations (GMACs), and the inference time for processing a single edited image. The best result for each metric is highlighted in bold. Our method incurs higher computational cost than prior encoder-based baselines. This overhead mainly stems from the additional discrepancy extraction and spatially adaptive fusion modules introduced to ensure detail consistency during editing, both of which require extra feature processing at inference time. Although these modules improve nonlinear and region-selective transformations while reducing misalignment artifacts, they also introduce a trade-off between editing quality and efficiency. Future work could explore lightweight designs or knowledge distillation strategies to reduce the computational overhead while maintaining high editing quality.

Table 5. Efficiency comparison in terms of parameters, computation, and inference time.

Method	Params (M)	GMACs	Time (s)
e4e	268.568	123.718	**0.036**
FS	**144.291**	**113.165**	0.042
HFGI	303.753	253.863	0.075
CLCAE	398.092	320.425	0.059
StyleRes	388.822	312.896	0.058
SDIC	321.002	269.421	0.113
SFE	424.601	380.121	0.078
Ours	502.738	689.103	0.121

Besides the increased computational overhead, our method still exhibits limitations under challenging editing scenarios involving large structural misalignment. As shown in Figure 10, when the target edit introduces substantial pose variation, the edited results may exhibit incomplete structural transformation and noticeable artifacts. We attribute this limitation to the fact that the extracted edit-discrepancy may be insufficient to capture complex geometric deformation, and the fusion process may therefore fail to fully align injected details with the edited semantic layout. Incorporating stronger geometric constraints into the training process may further improve robustness in such cases, and we leave this for future work.

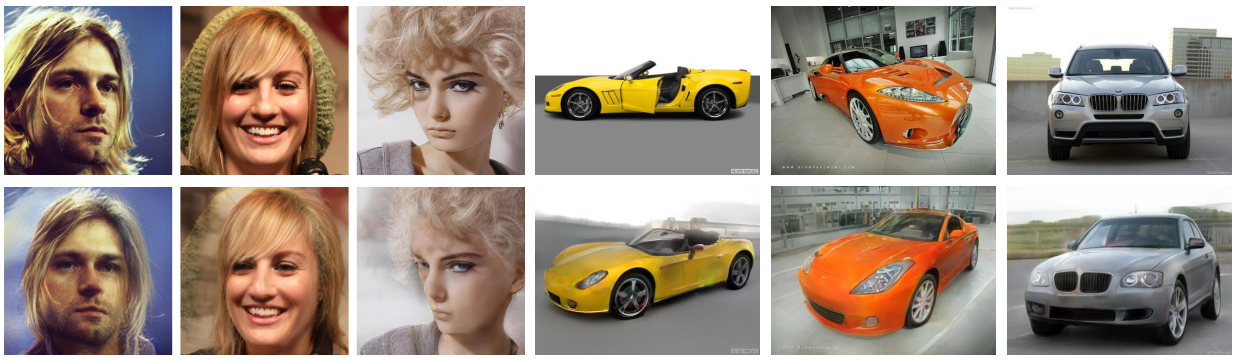

Figure 10. Representative failure cases under large-scale pose editing.

5. Conclusions

In this work, we revisited encoder-based GAN inversion from the perspective of how high-rate features should be handled during editing. Instead of applying global linear transformations to high-rate features (which can easily lead to misaligned details), we built a high-fidelity inversion backbone together with an adaptive feature editor that nonlinearly and locally transforms feature tensors under the guidance of edit-discrepancy signals. This design enables semantically consistent attribute edits while preserving fine details and reducing artifacts in the edited images. Extensive comparative experiments indicate that our approach achieves superior performance over existing encoder-based GAN inversion approaches in both reconstruction fidelity and editing quality. The contribution of each component is further verified by ablation studies.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province of China (No. 2026J001989), the Natural Science Foundation of Fujian Province of China (No. 2021J011007), Fujian Provincial Department of Education Undergraduate Education and Teaching Research Project (No. FBJY20230083), Principal's Foundation of Minnan Normal University (KJ19015), the Program for the Introduction of High-Level Talent of Zhangzhou, and the National Natural Science Foundation of China (No. 61702239).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Commun. ACM*, **63** (2020), 139–144. https://doi.org/10.1145/3422622
2. T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, preprint, arXiv:1710.10196. https://doi.org/10.48550/arXiv.1710.10196
3. T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, *IEEE Trans. Pattern Analy. Mach. Intell.*, **43** (2021), 4217–4228. https://doi.org/10.1109/TPAMI.2020.2970919
4. T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 8107–8116. https://doi.org/10.1109/CVPR42600.2020.00813

5. T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, in *Advances in Neural Information Processing Systems*, **33** (2020), 12104–12114.
6. T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, et al., Alias-free generative adversarial networks, in *Advances in Neural Information Processing Systems*, **34** (2021), 852–863.
7. R. Abdal, Y. Qin, P. Wonka, Image2stylegan++: How to edit the embedded images?, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 8296–8305. https://doi.org/10.1109/CVPR42600.2020.00832
8. J. Y. Zhu, P. Krähenbühl, E. Shechtman, A. A. Efros, Generative visual manipulation on the natural image manifold, in *European Conference on Computer Vision*, (2016), 597–613. https://doi.org/10.1007/978-3-319-46454-1_36
9. N. Tishby, N. Zaslavsky, Deep learning and the information bottleneck principle, in *2015 IEEE Information Theory Workshop*, (2015), 1–5. https://doi.org/10.1109/ITW.2015.7133169
10. X. Yao, A. Newson, Y. Gousseau, P. Hellier, A style-based gan encoder for high fidelity reconstruction of images and videos, in *European Conference on Computer Vision*, (2022), 581–597. https://doi.org/10.1007/978-3-031-19784-0_34
11. H. Liu, Y. Song, Q. Chen, Delving stylegan inversion for image editing: A foundation latent space viewpoint, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 10072–10082. https://doi.org/10.1109/CVPR52729.2023.00971
12. Z. Zhang, Y. Yan, J. H. Xue, H. Wang, Spatial-contextual discrepancy information compensation for GAN inversion, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2024), 7432–7440. https://doi.org/10.1609/aaai.v38i7.28574
13. T. Park, M. Y. Liu, T. C. Wang, J. Y. Zhu, Semantic image synthesis with spatially-adaptive normalization, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 2332–2341. https://doi.org/10.1109/CVPR.2019.00244
14. A. Creswell, A. A. Bharath, Inverting the generator of a generative adversarial network, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2019), 1967–1974. https://doi.org/10.1109/TNNLS.2018.2875194
15. R. Abdal, Y. Qin, P. Wonka, Image2stylegan: How to embed images into the stylegan latent space?, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 4431–4440. https://doi.org/10.1109/ICCV.2019.00453
16. P. Zhu, R. Abdal, J. Femiani, P. Wonka, Barbershop: GAN-based image compositing using segmentation masks, *ACM Trans. Graphics*, **40** (2021), 1–13. https://doi.org/10.1145/3478513.3480537
17. D. Roich, R. Mokady, A. H. Bermano, D. Cohen-Or, Pivotal tuning for latent-based editing of real images, *ACM Trans. Graphics*, **42** (2022), 1–13. https://doi.org/10.1145/3544777
18. E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, et al., Encoding in style: A stylegan encoder for image-to-image translation, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 2287–2296. https://doi.org/10.1109/CVPR46437.2021.00232

19. O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, D. Cohen-Or, Designing an encoder for stylegan image manipulation, *ACM Trans. Graphics*, **40** (2021), 1–14. https://doi.org/10.1145/3450626.3459838

20. Y. Alaluf, O. Patashnik, D. Cohen-Or, Restyle: A residual-based stylegan encoder via iterative refinement, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 6691–6700. https://doi.org/10.1109/ICCV48922.2021.00664

21. X. Hu, Q. Huang, Z. Shi, S. Li, C. Gao, L. Sun, et al., Style transformer for image inversion and editing, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 11327–11336. https://doi.org/10.1109/CVPR52688.2022.01105

22. H. Pehlivan, Y. Dalva, A. Dundar, Styleres: Transforming the residuals for real image editing with stylegan, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 1828–1837. https://doi.org/10.1109/CVPR52729.2023.00182

23. D. Bobkov, V. Titov, A. Alanov, D. Vetrov, The devil is in the details: Stylefeatureeditor for detail-rich stylegan inversion and high quality image editing, in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2024), 9337–9346. https://doi.org/10.1109/CVPR52733.2024.00892

24. A. B. Yildirim, H. Pehlivan, A. Dundar, Warping the residuals for image editing with stylegan, *Int. J. Comput. Vision*, **133** (2025), 2311–2326. https://doi.org/10.1007/s11263-024-02301-6

25. P. Isola, J. Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 5967–5976. https://doi.org/10.1109/CVPR.2017.632

26. J. Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 2242–2251. https://doi.org/10.1109/ICCV.2017.244

27. Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, J. Choo, Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 8789–8797. https://doi.org/10.1109/CVPR.2018.00916

28. Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, Attgan: Facial attribute editing by only changing what you want, *IEEE Trans. Image Process.*, **28** (2019), 5464–5478. https://doi.org/10.1109/TIP.2019.2916751

29. M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, et al., Stgan: A unified selective transfer network for arbitrary image attribute editing, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 3668–3677. https://doi.org/10.1109/CVPR.2019.00379

30. Z. Wu, D. Lischinski, E. Shechtman, Stylespace analysis: Disentangled controls for stylegan image generation, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 12858–12867. https://doi.org/10.1109/CVPR46437.2021.01267

31. Y. Shen, C. Yang, X. Tang, B. Zhou, Interfacegan: Interpreting the disentangled face representation learned by GANs, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2020), 2004–2018. https://doi.org/10.1109/TPAMI.2020.3034267

32. R. Abdal, P. Zhu, N. J. Mitra, P. Wonka, Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, *ACM Trans. Graphics*, **40** (2021), 1–21. https://doi.org/10.1145/3447648

33. E. Härkönen, A. Hertzmann, J. Lehtinen, S. Paris, Ganspace: Discovering interpretable GAN controls, in *Advances in Neural Information Processing Systems*, **33** (2020), 9841–9850.

34. O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, D. Lischinski, Styleclip: Text-driven manipulation of stylegan imagery, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 2065–2074. https://doi.org/10.1109/ICCV48922.2021.00209

35. J. Choi, Y. Choi, Y. Kim, J. Kim, S. Yoon, Custom-edit: Text-guided image editing with customized diffusion models, preprint, arXiv:2305.15779. https://doi.org/10.48550/arXiv.2305.15779

36. R. Jiang, X. Fu, G. Zheng, T. Li, T. Yao, X. Li, Energy-guided optimization for personalized image editing with pretrained text-to-image diffusion models, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2025), 4048–4056. https://doi.org/10.1609/aaai.v39i4.32424

37. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, et al., Swin transformer: Hierarchical vision transformer using shifted windows, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986

38. J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, R. Timofte, Swinir: Image restoration using swin transformer, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, (2021), 1833–1844. https://doi.org/10.1109/ICCVW54120.2021.00210

39. J. Krause, M. Stark, J. Deng, F. F. Li, 3D object representations for fine-grained categorization, in *2013 IEEE International Conference on Computer Vision Workshops*, (2013), 554–561. https://doi.org/10.1109/ICCVW.2013.77

40. T. Wang, Y. Zhang, Y. Fan, J. Wang, Q. Chen, High-fidelity gan inversion for image attribute editing, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 11369–11378. https://doi.org/10.1109/CVPR52688.2022.01109

41. Y. Alaluf, O. Tov, R. Mokady, R. Gal, A. Bermano, Hyperstyle: Stylegan inversion with hypernetworks for real image editing, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 18490–18500. https://doi.org/10.1109/CVPR52688.2022.01796

42. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 586–595. https://doi.org/10.1109/CVPR.2018.00068

43. Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, (2003), 1398–1402. https://doi.org/10.1109/ACSSC.2003.1292216

44. Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, et al., Curricularface: Adaptive curriculum learning loss for deep face recognition, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 5900–5909. https://doi.org/10.1109/CVPR42600.2020.00594

45. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in *Advances in Neural Information Processing Systems*, **30** (2017).

AIMS Press