DHFusion: Dual Dynamic Hypergraph-Driven Fusion for GAN Inversion

Kaitao Huang[1], Leying Jiang[1], Xiaokun Li[2], BoKai Ou[1], Xinyi Wang[1]

[1]Deep Learning AI Class
[2]Deep Learning Information School Class
{23020241154401, 36920241153221, 23020241154415, 36920241153243, 23020241154447}@stu.xmu.edu.cn

Abstract

Most existing GAN inversion methods aim to strike a good trade-off between fidelity and editability. However, a pre-trained GAN latent space only encodes information from in-domain regions while the input image often involves out-of-domain regions. As a result, extending the original latent space by encoding out-of-domain regions to improve fidelity will negatively affect the editability of the model. To address this, we propose a novel dual dynamic hypergraph-driven fusion (DHFusion) method, which consists of a dual dynamic hypergraph-CAM network (DH-Net) and an editing-driven fusion network (EF-Net). Specifically, DH-Net first employs the differential activations between the initial inverted image and the initial edited image to dynamically construct two hypergraphs from the perspectives of short-term and long-term spatial dependencies. In this way, high-order relationships between attribute-relevant regions are effectively modeled, enabling our model to generate an accurate and comprehensive edit-aware mask for locating the edited regions. Subsequently, EF-Net leverages this mask as weights to perform multi-scale feature-level fusion between the original image and the initial edited image, generating high-fidelity edited images with the reduced ghosting effect. Extensive quantitative and qualitative experiments demonstrate that our method outperforms several state-of-the-art methods. Our work clearly shows the potential of dual dynamic hypergraphs for GAN inversion.

Introduction

Over the past few years, image attribute editing, which aims to manipulate the desired attributes of an image, has received considerable attention. With the advance of generative adversarial networks (GANs) (Goodfellow et al. 2014), many efforts have been devoted to performing image attribute editing based on the controllability of powerful GAN models (such as StyleGAN (Karras, Laine, and Aila 2019; Karras et al. 2020)). Accordingly, a variety of GAN inversion methods (Xia et al. 2022) have been developed.

Traditional GAN inversion methods (Richardson et al. 2021; Tov et al. 2021) project the input images into the latent space of StyleGAN (i.e., the $\mathcal{W}$ space) and achieve good editing performance by varying the latent code. Unfortunately, these methods easily suffer from information loss due to the low bit-rate latent code. Thus, the fidelity of the reconstructed images is severely affected. To address this, some methods (Roich et al. 2022) fine-tune the generator. Although they can improve the inversion results of in-domain regions (such as facial and hair regions), the generator may fail to reconstruct out-of-domain regions (such as the complex background). A few methods (Wang et al. 2022; Zhang et al. 2024) incorporate the features extracted from the input image into the generator, enhancing the capability of pre-trained GAN models for out-of-domain inversion. Such a way improves the reconstruction quality of the edited image but sacrifices the editability. Thus, these methods still cannot achieve a good trade-off between fidelity and editability.

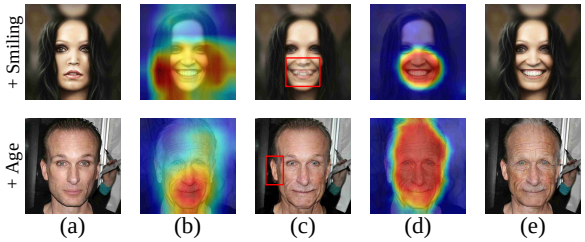

Figure 1: Visualization of (a) the original images; (b) the masks generated by Diff-CAM; (c) the final edited results by Diff-CAM; (d) the masks generated by our method; (e) the final edited results by our method.

Recently, some methods (such as Diff-CAM (Song et al. 2022)) have been proposed to combine the edited regions from the inversion with the unedited regions from the original input. They introduce a differential activation module to train an attribute classifier and generate a mask for localizing edited regions. Based on the generated mask, the original image and the initial edited image are directly fused as the edited result.

The above methods mainly suffer from two main drawbacks. First, they perform attribute classification and obtain the activation map by using Grad-CAM (Selvaraju et al. 2017). Nevertheless, these methods do not fully exploit the relationships between different attribute-relevant regions and thus they can easily generate inaccurate or incomplete masks, leading to unsatisfactory editing results. Second, by blending the original and edited images with the generated mask at the pixel level, these methods often cause ghosting

effects, even with the employment of a deghosting network. Some failed cases are shown in Figures 1(b) and 1(c).

To address the above problems, in this paper, we propose a novel dual dynamic hypergraph-driven fusion (DHFusion) method for GAN inversion. DHFusion mainly consists of a dual dynamic hypergraph-CAM network (DH-Net) and an editing-driven fusion network (EF-Net) for accurate and comprehensive edited region localization and high-quality edited image generation with the reduced ghosting effect, respectively. Specifically, DH-Net dynamically constructs two hypergraphs to capture high-order correlations from the perspectives of short-term and long-term spatial dependencies and generates an edit-aware mask. Based on the generated mask, EF-Net performs multi-scale feature-level fusion between the original image and the initial edited image and offers multi-scale fused feature maps. This can facilitate the generation of a high-fidelity edited image. Some generated masks and edited results by our method are given in Figures 1(d) and 1(e), respectively.

In summary, the contributions of this paper are as follows:

- We propose a novel method for GAN inversion, which leverages an edit-aware mask to fuse the original image and the initial edited image at the feature level, achieving an excellent trade-off between fidelity and editability.
- We propose DH-Net based on dual dynamic hypergraph construction, generating an accurate and comprehensive edit-aware mask. Moreover, we design EF-Net to perform multi-scale feature-level fusion between the original image and the initial edited image. In this way, our method can effectively preserve out-of-domain regions while achieving high editing quality with a reduced ghosting effect.
- Qualitative and quantitative experiments validate the superiority of our method against several state-of-the-art GAN inversion methods.

Related Work

GAN Inversion. Existing GAN inversion methods can be roughly divided into three categories: optimization-based, encoder-based, and hybrid methods. Optimization-based methods (Abdal, Qin, and Wonka 2020; Zhu et al. 2020b) directly optimize the latent code by minimizing the reconstruction loss for each image. Although these methods can reconstruct high-fidelity images, they easily suffer from poor editability and slow inference time. Encoder-based methods (Tov et al. 2021; Richardson et al. 2021) train an encoder to map the input images into the latent space. Thus, they can perform attribute editing operations in the latent space. Compared with optimization-based methods, encoder-based methods offer better editability and faster inference time. But their reconstruction quality may be poor. Hybrid methods (Zhu et al. 2016, 2020a) first utilize an encoder to obtain a latent code and then optimize this latent code. They can achieve a good balance between inference time and reconstruction quality. Recently, some methods (Alaluf et al. 2022) make use of the hypernetwork to calculate changes in the weights of the GAN generator, improving the reconstruction quality.

Although the above methods have progressed greatly, they still struggle to invert out-of-domain regions. Recently, Diff-CAM (Song et al. 2022) proposes localizing the edited region with a mask and blending them with the original image, to improve image fidelity. However, the localization capability of Diff-CAM is limited, generating inaccurate or incomplete masks. Moreover, the ghosting effect caused by pixel-level blending still exists, even with the adoption of a deghosting network. SAMM (Yang, Xu, and Chen 2023) applies spatial alignment to reduce the ghosting effect. However, the generated pixel-level mask may be inaccurate when dealing with complex out-of-domain regions.

Hypergraph Learning. Recently, hypergraph neural network-based methods (Wadhwa et al. 2021; Han et al. 2023) have made great progress in computer vision, where their performance relies heavily on the quality of the constructed hypergraph structures. DHGNN (Jiang et al. 2019) introduces dynamic hypergraph construction using K-means and K nearest neighbors (KNN). ViHGNN (Han et al. 2023) alternately performs patch embeddings and hypergraph construction, enhancing structure-aware image representations. Some methods (Wadhwa et al. 2021) employ cross-correlation between vertex features to learn the incidence matrix.

Unlike the conventional hypergraph construction methods that model the data structure from either short-term or long-term dependencies, we introduce dual dynamic hypergraph construction to combine short-term and long-term dependencies. Notably, we progressively update vertex features and the dual hypergraph structure in an alternate learning way. Such a manner is beneficial for sufficiently establishing high-order relationships between attribute-relevant regions.

Proposed Solution

Overview

The overview of our dual dynamic hypergraph-driven fusion (DHFusion) method is shown in Figure 2. DHFusion consists of DH-Net and EF-Net, which are trained separately. DH-Net, consisting of a differential activation-based hypergraph learning (DHL) module and an edit-aware attention (EA) module, is trained to generate an edit-aware mask. EF-Net is trained to give the final edited result via multi-scale feature-level fusion.

Dual Dynamic Hypergraph-CAM Network

The DHL Module The DHL module is designed to extract the relational features by exploiting high-order correlations between different attribute-relevant regions. To achieve this, we dynamically construct dual hypergraphs from the perspectives of short-term and long-term spatial dependencies. The DHL module includes three key components: 1) differential activations; 2) dual hypergraph construction; and 3) dual dynamic hypergraph learning.

Differential Activations. We first employ an editing method (such as pSp (Richardson et al. 2021), e4e (Tov et al. 2021)) to perform the initial inversion and editing, obtaining an initial inverted image $\mathbf{I}'$ and an initial edited image $\mathbf{T}$. Then we

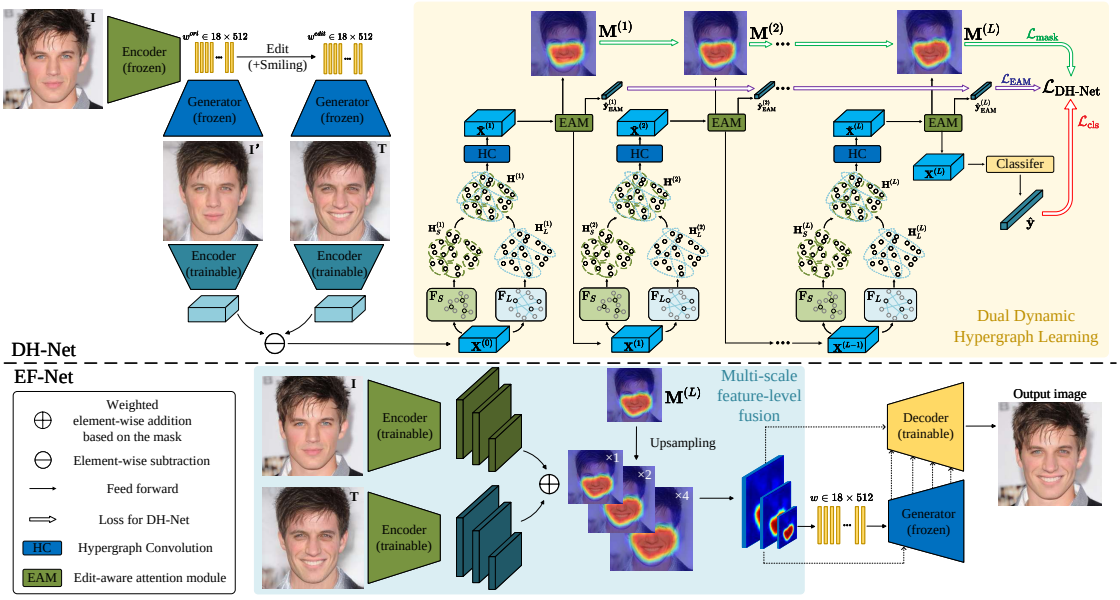

Figure 2: Overview of our DHFusion. DHFusion mainly consists of a dual dynamic hypergraph-CAM network (DH-Net) and an editing-driven fusion network (EF-Net).

compute the differential activations $\mathbf{\Delta}$ (Song et al. 2022) as

$$\mathbf{\Delta} = \mathrm{E}_{\text{trainable}}(\mathbf{I}') - \mathrm{E}_{\text{trainable}}(\mathbf{T}), \tag{1}$$

where $\mathrm{E}_{\text{trainable}}(\cdot)$ is a plain trainable encoder and $\mathbf{\Delta} \in \mathbb{R}^{C\times H\times W}$. Here, C, H, and W denote the channel number, height, and width of $\mathbf{\Delta}$, respectively.

Dual Hypergraph Construction. We consider each $1\times 1\times C$ grid of the differential activations $\mathbf{\Delta}$ as an initial vertex in each hypergraph. Thus, we can transform $\mathbf{\Delta}$ into the initial relational features (vertex features) $\mathbf{X}^{(0)} \in \mathbb{R}^{C\times HW}$, where the n-th vertex is represented by a feature $\mathbf{x}_n^{(0)} \in \mathbb{R}^{C\times 1}$ and HW denotes the total number of vertices.

Short-Term Spatial Dependencies. We construct the short-term incidence matrix $\mathbf{H}_S$ based on K-nearest neighbors. Technically, we first use the KNN algorithm to select the K nearest vertices to each vertex (using the cosine distance) and filter out neighbors whose distances are greater than a threshold for each vertex, obtaining the short-term incidence matrix.

Mathematically, the initial short-term incidence matrix $\mathbf{H}_S^{(1)} \in \mathbb{R}^{HW\times HW}$ is constructed as

$$\mathbf{H}_S^{(1)} = \mathrm{F}_S(\mathbf{X}^{(0)}, K, \epsilon), \tag{2}$$

where ϵ is the threshold to filter out neighbors with large distances and $\mathrm{F}_S(\cdot)$ denotes the short-term hypergraph construction.

Long-Term Spatial Dependencies. We construct the long-term incidence matrix $\mathbf{H}_L$ using cross-correlation (Wadhwa et al. 2021) between relational features, which measures the contribution of each vertex in each hyperedge, that is,

$$\begin{aligned}\mathbf{H}_L^{(1)} &= \mathrm{F}_L(\mathbf{X}^{(0)}) \\ &= \Psi(\mathbf{X}^{(0)})\Lambda(\mathbf{X}^{(0)})\Psi(\mathbf{X}^{(0)})^{\mathrm{T}}\Omega(\mathbf{X}^{(0)}),\end{aligned} \tag{3}$$

where $\mathbf{H}_L^{(1)} \in \mathbb{R}^{HW\times HW}$ denotes the initial long-term incident matrix (we set the number of hyperedges to HW); $\mathrm{F}_L(\cdot)$ denotes the long-term hypergraph construction; $\Psi(\cdot)$ denotes a linear transformation; $\Lambda(\cdot)$ and $\Omega(\cdot)$ are learnable parameters, which learn a distance metric between the vertices for the incidence matrix and determine the contribution of each vertex on the hyperedge, respectively.

Dual Dynamic Hypergraph Learning. At the l-th iteration, the relational features $\mathbf{X}^{(l-1)} \in \mathbb{R}^{C\times HW}$ from the $(l-1)$-th iteration are used as the input for both short-term and long-term hypergraph construction. Hence, the incidence matrices for the short-term and long-term hypergraphs can be obtained as

$$\mathbf{H}_S^{(l)} = \mathrm{F}_S(\mathbf{X}^{(l-1)}, K, \epsilon), \tag{4}$$

$$\mathbf{H}_L^{(l)} = \mathrm{F}_L(\mathbf{X}^{(l-1)}). \tag{5}$$

Then, we add $\mathbf{H}_S^{(l)}$ and $\mathbf{H}_L^{(l)}$ to obtain the fused incidence matrix $\mathbf{H}^{(l)}$.

Based on $\mathbf{H}^{(l)} \in \mathbb{R}^{HW\times HW}$, we apply a general hypergraph convolutional layer (Feng et al. 2019) to aggregate high-order structure information and then enhance relational feature representations. The enhanced relational features $\tilde{\mathbf{X}}^{(l)} \in \mathbb{R}^{C\times HW}$ can be obtained as

$$\begin{aligned}\tilde{\mathbf{X}}^{(l)} &= \mathrm{HC}(\mathbf{X}^{(l-1)}, \mathbf{H}^{(l)}) \\ &= \sigma(\mathbf{D}_v^{-1/2}\mathbf{H}^{(l)}\mathbf{W}\mathbf{D}_e^{-1}(\mathbf{H}^{(l)})^{\mathrm{T}}\mathbf{D}_v^{-1/2}\mathbf{X}^{(l-1)}\mathbf{\Theta}),\end{aligned} \tag{6}$$

where $\mathbf{D}_v$, $\mathbf{D}_e$, and $\mathbf{W}$ represent the diagonal matrices of vertex degrees, edge degrees, and edge weights, respectively; $\mathbf{\Theta}$ denotes the learnable parameters of the hypergraph convolutional layer that are shared during each iteration; σ is an activation function (we use an exponential linear unit (ELU) (Clevert, Unterthiner, and Hochreiter 2020)); $\mathrm{HC}(\cdot)$ denotes the hypergraph convolutional operation.

At the end of each iteration, we leverage an edit-aware attention (EA) module (see details in next Section) to compute an edit-aware mask $\mathbf{M}^{(l)}$ and output the edit-aware relational features $\mathbf{X}^{(l)}$.

To train the model, we append a classifier (a global average pooling (GAP) layer, followed by a fully-connected (FC) layer and a softmax function) after the output of the last iteration $\mathbf{X}^{(L)}$ (L denotes the total number of iterations), and generate a vector $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_c\}$ that indicates the classification probabilities for each editing attribute, where c is the number of attributes.

Edit-Aware Attention (EA) Module Inspired by ABN (Fukui et al. 2019), we design an EA module to enable the model to focus on the edited regions and generate an edit-aware mask. The EA module takes the enhanced relational features $\tilde{\mathbf{X}}^{(l)} \in \mathbb{R}^{C\times HW}$ as the input and outputs the edit-aware relational features $\mathbf{X}^{(l)} \in \mathbb{R}^{C\times HW}$, an edit-aware mask $\mathbf{M}^{(l)} \in \mathbb{R}^{H\times W}$, and an output vector $\hat{\mathbf{y}}^{(l)}_{\text{EAM}} \in \mathbb{R}^c$ of a classifier. Note that, at each iteration, the input features $\tilde{\mathbf{X}}^{(l)}$ are transformed into the spatial features $\tilde{\mathbf{F}}^{(l)} \in \mathbb{R}^{C\times H\times W}$ and the edit-aware spatial features $\mathbf{F}^{(l)} \in \mathbb{R}^{C\times H\times W}$ are transformed back into the edit-aware relational features $\mathbf{X}^{(l)}$ as the output.

For the input spatial features $\tilde{\mathbf{F}}^{(l)}$, we first use a self-attention block (Vaswani et al. 2017) to capture potential relationships between each position comprehensively. The output of the self-attention block is denoted as $\mathbf{F}^{(l)}_{\text{self}} \in \mathbb{R}^{C\times H\times W}$. Then, $\mathbf{F}^{(l)}_{\text{self}}$ is further fed into two branches: the attention branch and the classification branch.

For the attention branch, a convolutional layer is used to adjust the number of channels of $\mathbf{F}^{(l)}_{\text{self}}$ to 1, followed by a Sigmoid activation function, obtaining spatial attention weights $\mathbf{W}^{(l)}_S \in \mathbb{R}^{1\times H\times W}$. For the classification branch, a classifier takes $\mathbf{F}^{(l)}_{\text{self}}$ as the input and generates the classification prediction vector $\hat{\mathbf{y}}^{(l)}_{\text{EAM}}$.

Based on the above, $\mathbf{W}^{(l)}_S$ can be used to highlight the edited regions of the spatial features:

$$\mathbf{F}^{(l)} = \tilde{\mathbf{F}}^{(l)} + \mathbf{W}^{(l)}_S \odot \tilde{\mathbf{F}}^{(l)}. \tag{7}$$

where '$\odot$' denotes the Hadamard product.

Meanwhile, we perform min-max normalization on the spatial weights $\mathbf{W}^{(l)}_S$ to obtain an edit-aware mask $\mathbf{M}^{(l)}$. During the model inference stage, we use $\mathbf{M}^{(L)}$ obtained from the last iteration as the final edit-aware mask predicted by DH-Net.

Loss Function of DH-Net The loss function for optimizing DH-Net is defined as

$$\mathcal{L}_{\text{DH-Net}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{EAM}}\mathcal{L}_{\text{EAM}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}}, \tag{8}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{EAM}}$ are cross-entropy losses, which enable the model to generate edit-aware masks by training classifiers; $\mathcal{L}_{\text{mask}}$ is the L_1 loss that prevents the size of the generated edit-aware mask from excessive coverage of out-of-domain regions; λ_{EAM} and λ_{mask} are the balancing parameters.

Editing-Driven Fusion Network

As illustrated in Figure 2, our EF-Net consists of an encoder, a decoder, and a pre-trained StyleGAN2 generator. First, we input the original image $\mathbf{I}$ and the initial edited image $\mathbf{T}$ into the encoder to obtain three different scales of encoded features (including coarse, medium, and fine features). Then, $\mathbf{M}^{(L)}$ is upsampled to three different scales $\{\mathbf{M}_i\}^3_{i=1}$ (i.e., 16×16, 32×32, and 64×64), corresponding to different feature scales respectively. Hence, the fused features can be obtained as

$$\mathbf{F}^{\text{fusion}}_i = \mathbf{F}^T_i \odot \mathbf{M}_i + \mathbf{F}^I_i \odot (1 - \mathbf{M}_i), i = 1, 2, 3 \tag{9}$$

where $\mathbf{F}^{\text{fusion}}_1$, $\mathbf{F}^{\text{fusion}}_2$, and $\mathbf{F}^{\text{fusion}}_3$ denote the fused features at the coarse, medium, and fine levels, respectively; $\mathbf{F}^I_i$ and $\mathbf{F}^T_i$ denote the encoded features for the original image and the initial edited image at one scale, respectively.

Subsequently, we use a decoder to decode the fused features into the final edited image. Meanwhile, $\{\mathbf{F}^{\text{fusion}}_i\}^3_{i=1}$ is mapped to latent code $w \in \mathbb{R}^{18\times 512}$, which are then fed into the generator to generate ghosting-free features. We aggregate the features from the generator with those from the decoder in a hierarchical manner to give the final results.

Loss Function of EF-Net. To encourage EF-Net to have the ability of deghosting, inspired by Diff-CAM, we generate a set of paired data $\{\mathbf{I}_{\text{train}}, \mathbf{I}\}$, where $\mathbf{I}_{\text{train}}$ is obtained by

$$\mathbf{I}_{\text{train}} = \mathbf{T} \odot \mathbf{M}_{\text{train}} + \mathbf{I} \odot (1 - \mathbf{M}_{\text{train}}), \tag{10}$$

and each element in $\mathbf{M}_{\text{train}}$ is computed by

$$\mathbf{M}_{\text{train}}(i,j) = \begin{cases} \mathbf{M}^{(L)}(i,j) & \text{if } \mathbf{M}^{(L)}(i,j) \le 0.5 \\ 1 - \mathbf{M}^{(L)}(i,j) & \text{if } \mathbf{M}^{(L)}(i,j) > 0.5, \end{cases} \tag{11}$$

During training, EF-Net solely takes $\mathbf{I}_{\text{train}}$ as the input and generates the output image $\mathbf{I}_{\text{rec}}$ through the decoder, while the generator is frozen. In this way, EF-Net can be trained by using $\mathbf{I}$ as the ground truth, enabling the model to have the ability to remove ghosting.

To obtain a high-fidelity edited image after feature fusion, we compute the L_2 loss ($\mathcal{L}_2$) and the LPIPS loss ($\mathcal{L}_{\text{LPIPS}}$) (Zhang et al. 2018) between $\mathbf{I}_{\text{rec}}$ and $\mathbf{I}$. In addition, we calculate the identity loss $\mathcal{L}_{\text{id}} = 1 - \langle \text{F}(\mathbf{I}_{\text{rec}}), \text{F}(\mathbf{I}) \rangle$, where $\text{F}(\cdot)$ denotes a pre-trained ArcFace (Deng et al. 2019) for the face domain and a pre-trained ResNet-50 model (Tov et al. 2021) for other domains. Thus, the loss function of EF-Net is

$$\mathcal{L}_{\text{EF-Net}} = \mathcal{L}_2 + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{id}}\mathcal{L}_{\text{id}}, \tag{12}$$

where λ_{LPIPS} and λ_{id} are the balancing parameters.

Experiments

Experimental Settings

Datesets. We mainly evaluate our method on the face domain. We adopt the FFHQ dataset (Karras, Laine, and Aila 2019) for training and the CelebA-HQ dataset (Karras et al. 2017) for testing. The training set and the test set contain 70,000 and 30,000 human facial images, respectively. Each image has the size of 1024×1024.

Figure 3: Image attribute editing quality comparison between DHFusion and several state-of-the-art methods.

Implementation Details. For short-term hypergraph construction, the values of K and ϵ are set to 25 and 0.5, respectively. The number of iterations L is set to 3. The values of λ_{EAM} and λ_{mask} in Eq. (8) are empirically set to 0.1 and 0.01, respectively. The values of λ_{LPIPS} and λ_{id} in Eq. (12) are empirically set to 0.8 and 0.1, respectively. The input size and output size of EF-Net are 256×256 and 1024×1024, respectively.

To evaluate the realism of the generated images, we use Fréchet Inception Distance (FID) (Heusel et al. 2017) to measure the distribution distance between the original image dataset and the edited dataset generated by each model, where five facial editing attributes ("Beard", "Bushy eyebrows", "Eyeglasses", "Age", and "Smiling") are used to take the average. We also use ArcFace (Deng et al. 2019) to extract the features from the original image and the edited image and calculate their cosine similarity as the identity similarity (ID). The above metrics are evaluated on the first 1,000 images of CelebA-HQ.

Comparison with State-of-the-Art Methods

To show the superiority of our method, we compare our DHFusion with state-of-the-art GAN inversion methods, including pSp (Richardson et al. 2021), e4e (Tov et al. 2021), HFGI (Wang et al. 2022), HyperStyle (Alaluf et al. 2022), Diff-CAM (Song et al. 2022), SAMM (Yang, Xu, and Chen 2023), and SDIC (Zhang et al. 2024).

Quantitative Evaluation. We quantitatively compare our method with state-of-the-art GAN inversion methods using FID and ID as evaluation metrics. Meanwhile, we conduct a User Study to evaluate the editing results (Zhang et al. 2024). In addition, we compare the inference time obtained by different methods. The results are shown in Table 1. Among all the competing methods, our method obtains the best FID, ID, and User Study performance at the comparable inference time.

Table 1: Comparison with state-of-the-art methods.

Method	FID ↓	ID (%) ↑	Time (s) ↓	User Study (%) ↑
pSp	50.878	79.783	0.090	28.668
e4e	48.782	78.146	0.087	22.000
HFGI	39.524	83.822	0.175	8.666
HyperStyle	39.095	80.450	0.280	16.666
Diff-CAM	26.656	91.841	0.244	47.334
SAMM	32.267	87.040	0.186	14.666
SDIC	35.521	88.508	0.280	21.998
Ours	**18.875**	**98.803**	0.253	**59.334**

Qualitative Evaluation. As shown in Figure 3, we can observe that, compared with other methods, mask-based methods (i.e., Diff-CAM, SAMM, and our DHFusion) more effectively preserve out-of-domain regions when editing faces. In comparison to Diff-CAM, the edited images generated by our model show better editing performance. This is because our method can more accurately and comprehensively locate the edited region and greatly alleviate the ghosting effect during image editing.

Conclusions

In this paper, we propose a novel DHFusion method for GAN inversion. Our method is comprised of DH-Net and EF-Net. DH-Net first generates an accurate and comprehensive edit-aware mask to indicate the edited regions based on the differential activations between the initial inverted image and the initial edited image. Based on the generated mask, EF-Net then blends the original image with the initial edited image at the multi-scale feature level. Experimental results demonstrate that our method gives a final edited image that preserves out-of-domain regions while maintaining high editing quality with reduced ghosting effect. This shows our method achieves a favorable balance between fidelity and editability.

References

Abdal, R.; Qin, Y.; and Wonka, P. 2020. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8296–8305.

Alaluf, Y.; Tov, O.; Mokady, R.; Gal, R.; and Bermano, A. 2022. Hyperstyle: StyleGAN inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18511–18521.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2020. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv 2015. *arXiv preprint arXiv:1511.07289*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3558–3565.

Fukui, H.; Hirakawa, T.; Yamashita, T.; and Fujiyoshi, H. 2019. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10705–10714.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*.

Han, Y.; Wang, P.; Kundu, S.; Ding, Y.; and Wang, Z. 2023. Vision HGNN: An Image is more than a graph of nodes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19878–19888.

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

Jiang, J.; Wei, Y.; Feng, Y.; Cao, J.; and Gao, Y. 2019. Dynamic hypergraph neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2635–2641.

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

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–8119.

Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; and Cohen-Or, D. 2021. Encoding in style: a StyleGAN encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2287–2296.

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

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626.

Song, H.; Du, Y.; Xiang, T.; Dong, J.; Qin, J.; and He, S. 2022. Editing out-of-domain GAN inversion via differential activations. In *Proceedings of European Conference on Computer Vision*, 1–17.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wadhwa, G.; Dhall, A.; Murala, S.; and Tariq, U. 2021. Hyperrealistic image inpainting with hypergraphs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3912–3921.

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

Xia, W.; Zhang, Y.; Yang, Y.; Xue, J.-H.; Zhou, B.; and Yang, M.-H. 2022. GAN inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3121–3138.

Yang, X.; Xu, X.; and Chen, Y. 2023. Out-of-domain GAN inversion via invertibility decomposition for photo-realistic human face manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7492–7501.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 586–595.

Zhang, Z.; Yan, Y.; Xue, J.-H.; and Wang, H. 2024. Spatial-contextual discrepancy information compensation for GAN inversion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7432–7440.

Zhu, J.; Shen, Y.; Zhao, D.; and Zhou, B. 2020a. In-domain GAN inversion for real image editing. In *Proceedings of European Conference on Computer Vision*, 592–608.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision*, 597–613.

Zhu, P.; Abdal, R.; Qin, Y.; Femiani, J.; and Wonka, P. 2020b. Improved StyleGAN embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*.