HyperEditor: Achieving Both Authenticity and Cross-Domain Capability in Image Editing via Hypernetworks

Hai Zhang[1,2], **Chunwei Wu**[1,2], **Guitao Cao**[1,2*], **Hailing Wang**[1,2], **Wenming Cao**[3]

[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
[2]MoE Engineering Research Center of SW/HW Co-design Technology and Application, East China Normal University, China
[3]College of Information Engineering, Shenzhen University, China
{hzhang22, 52215902005}@stu.ecnu.edu.cn, gtcao@sei.ecnu.edu.cn, 52215902004@stu.ecnu.edu.cn, wmcao@szu.edu.cn

Abstract

Editing real images authentically while also achieving cross-domain editing remains a challenge. Recent studies have focused on converting real images into latent codes and accomplishing image editing by manipulating these codes. However, merely manipulating the latent codes would constrain the edited images to the generator's image domain, hindering the attainment of diverse editing goals. In response, we propose an innovative image editing method called HyperEditor, which utilizes weight factors generated by hypernetworks to reassign the weights of the pre-trained StyleGAN2's generator. Guided by CLIP's cross-modal image-text semantic alignment, this innovative approach enables us to simultaneously accomplish authentic attribute editing and cross-domain style transfer, a capability not realized in previous methods. Additionally, we ascertain that modifying only the weights of specific layers in the generator can yield an equivalent editing result. Therefore, we introduce an adaptive layer selector, enabling our hypernetworks to autonomously identify the layers requiring output weight factors, which can further improve our hypernetworks' efficiency. Extensive experiments on abundant challenging datasets demonstrate the effectiveness of our method.

Introduction

The primary objective of image editing is to modify specific properties of real images by leveraging conditions. In recent years, image editing has emerged as one of the most dynamic and vibrant research areas in academia and industry. Several studies (Shen et al. 2019; Härkönen et al. 2020; Liu et al. 2023; Revanur et al. 2023; Liu, Song, and Chen 2023; Patashnik et al. 2021) have investigated the extensive latent semantic representations in StyleGAN2 (Karras et al. 2020) and have successfully achieved diverse and authentic image editing through the manipulation of latent codes. These methods share a common characteristic: finding the optimal target latent codes for substituting the initial latent codes, thereby advancing the source image to the target image. However, latent codes with high reconstructability often exhibit weak editability (Pehlivan, Dalva, and Dundar 2023). Additionally, if the target image falls outside the image domain of the generator, it becomes difficult to achieve cross-domain image editing purely through the manipulation of the latent codes. Thus, we pondered the question: *"Is it possible to achieve both authentic image attribute editing and cross-domain image style transfer simultaneously?"*

Recently, (Alaluf et al. 2022; Dinh et al. 2022) utilize hypernetworks to reassign the generator's weights, achieving a more precise image reconstruction by gradually compensating for the missing details of the source image in the reconstructed image. Inspired by this, we find that modulating the generator's weights can achieve detailed attribute changes in images. Therefore, we adopt the concept of weight reassignment to the image editing task. In our work, we propose a novel image editing method called HyperEditor, which directly conducts image editing by utilizing weight factors generated by hypernetworks to reassign the weights of StyleGAN2's generator. Unlike traditional methods of model weight fine-tuning (Gal et al. 2022), which often involve retraining the pre-trained model, our approach involves scaling and reassigning the weights of the generator using weight factors. As a result, our method offers better controllability, allowing for authentic attribute editing (e.g., facial features, hair, etc.). Moreover, due to our method's capability to modify the generator's weights, it can effectively perform cross-domain editing operations, which might be challenging to achieve solely by manipulating the latent codes. To the best of our knowledge, our method is the first to achieve authentic attribute editing and cross-domain style editing simultaneously.

We pondered a question: must we reassign all layers in StyleGAN2's generator when editing a single attribute? Based on experimental findings, we observed that only a few layers' weight factors undergo significant changes before and after editing. Thus, we propose the adaptive layer selector, enabling hypernetworks to choose the layers that require outputting weight factors autonomously. Consequently, we can maximize the effect of weight factors while achieving comparable results.

During model training, aligning the generated images with the target conditions poses a challenge due to the lack of paired datasets before and after editing in the real-world. Recently, some methods (Wei et al. 2022; Kocasari et al. 2022) use CLIP (Radford et al. 2021) to convert image features into pseudo-text features and align them with genuine-text features. In our work, we leverage the self-supervised

*Corresponding author.

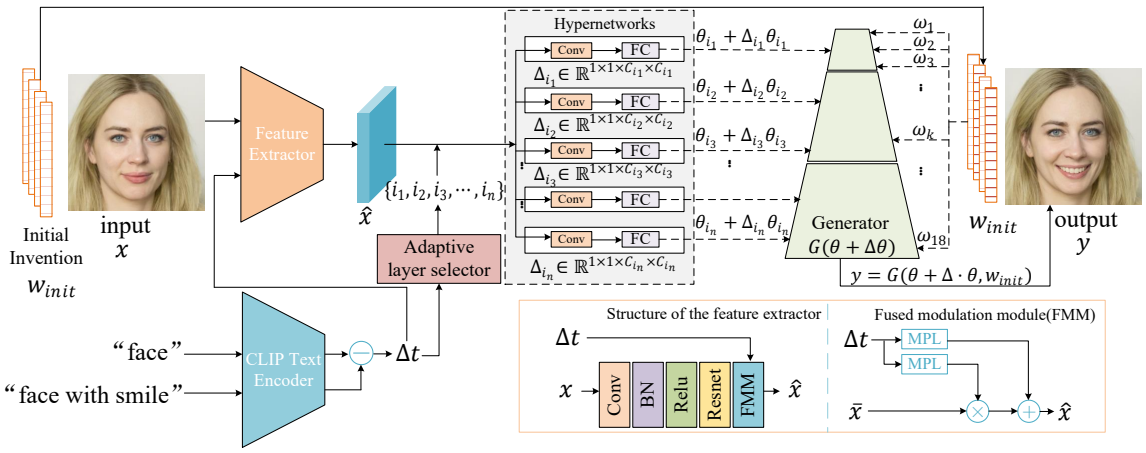

Figure 1: The overall structure of our HyperEditor. Given a text pair (t_0, t_1) and an initial image x, we utilize the CLIP text encoder to extract features for (t_0, t_1) and compute their difference, resulting in Δt. The image feature extractor processes x to obtain its feature representation, which is then refined with the conditional information Δt through the fusion modulation module(FMM), yielding an intermediate feature map, $\hat{x}$. Using the adaptive layer selector, we identify a sequence of layers, L, that require outputting weight factors. Subsequently, our hypernetworks generate weight factors, Δ, based on $\hat{x}$ and L, which are used to reassign the generator's weights. Moreover, we input the latent codes w_{init} of the initial image. Finally, the generated image after editing, $y = (\theta + \Delta \cdot \theta, w_{init})$, is obtained.

learning capacity of CLIP and introduce the directional CLIP loss to supervise model training. This process aligns the difference set of pseudo-text features before and after editing with the difference set of authentic-text pair features, all in a direction-based manner. It makes our approach more focused on cross-modal representations of local semantics, enhances the convergence capability of the model, and prevents the generation of adversarial effects. Additionally, we introduce a fusion modulation module that refines intermediate feature maps using text prompts as scaling and shifting factors. This enables different text prompts to manipulate the hypernetworks and generate various weight factors. Consequently, our approach enables a single model to accomplish diverse image editing tasks. Overall, our contributions can be summarized as follows:

- We surpass the constraints of prior image editing techniques and introduce a novel image editing framework called HyperEditor. This framework utilizes hypernetworks to reassign the weights of StyleGAN2's generator and leverages CLIP's cross-modal semantic alignment ability to supervise our model's training. Consequently, HyperEditor can not only authentically modify attributes in images but also achieve cross-domain style editing.
- We propose an adaptive layer selector that allows hypernetworks to autonomously determine the layers that require outputting weight factors when editing a single attribute, maximizing the efficiency of the hypernetworks.

Related Work

Image Editing

Many studies (Saha et al. 2021; Zhu et al. 2022; Roich et al. 2022) have investigated how to leverage the latent space of pre-trained generators for image editing. Particularly, with the advent of StyleGAN2, image editing through manipulation of the latent codes has become a prevalent research topic. In (Shen et al. 2019), the authors utilized linear variation to achieve a disentangled representation of the latent codes in StyleGAN. They completed accurate image editing by decoupling entangled semantics and subspace projection. In (Härkönen et al. 2020), the authors proposed using principal component analysis to identify and modify the latent directions in latent codes, thereby enabling image editing. TediGAN (Xia et al. 2021) introduced a visual-language similarity module that maps linguistic representations into a latent space that aligns with visual representations, allowing text-guided image editing. StyleCLIP (Patashnik et al. 2021) combines the robust cross-modal semantic alignment capability of CLIP with the generative power of StyleGAN2. It presents three text-driven image editing methods, namely latent optimization, latent mapping, and global directions, to achieve image editing in an unsupervised or self-supervised manner. Subsequently, a series of CLIP+StyleGAN2 methods (Wei et al. 2022; Lyu et al. 2023) have been introduced. These methods utilize mappers to learn style residuals and transform the original image's latent codes toward the target latent codes. Furthermore, several studies (Zeng, Lin, and Patel 2022; Revanur et al. 2023; Hou et al. 2022) have incorporated mask graphs into the network to provide more accurate supervision for specific attribute changes. We can observe that all these methods involve the manipulation of latent codes. However, if the target image exceeds the image domain of the generator, it becomes challenging to rely solely on modifying latent codes to achieve cross-domain editing operations. Thus, we take an approach, focusing on the generator's weights and performing image editing by reassigning these weights. Our method is entirely distinct from theirs, as it does not involve manipulating latent codes dur-

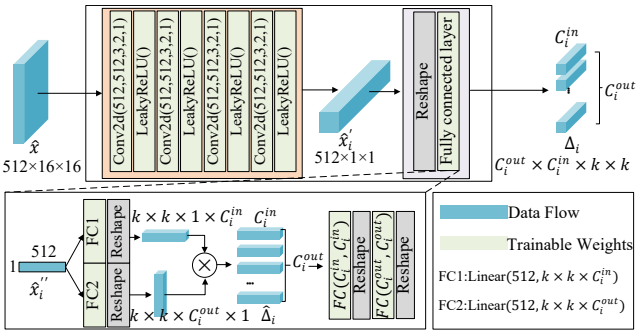

Figure 2: The structure of a hypernetwork. It consists of downsampled convolutions and fully connected layers. The hypernetwork takes the intermediate feature map $\hat{x}$ as input, which predicts and outputs the weight factor Δ_i for the convolutional layer i in StyleGAN2's generator.

ing image editing. Our approach sets the groundwork for developing novel image editing techniques.

Hypernetworks

A hypernetwork is an auxiliary neural network responsible for generating weights for another network, often called the primary network. It was initially introduced by (Ha, Dai, and Le 2017). Training the hypernetworks on extensive datasets can adjust the weights of the main network through appropriate weights shifts, resulting in a more expressive model. Since its proposal, hypernetworks have found applications in various domains, including semantic segmentation (Nirkin, Wolf, and Hassner 2021), neural architecture search (Zhang, Ren, and Urtasun 2019), 3D modeling (Littwin and Wolf 2019; Sitzmann et al. 2020), continuous learning (von Oswald et al. 2020), and more. Recently, hypernetworks have also been applied to the StyleGAN inversion task. Both (Alaluf et al. 2022) and (Dinh et al. 2022) have developed hypernetworks structures to enhance the quality of image reconstruction. However, they employ additional methods (such as StyleCLIP, etc.) to complete image editing. In contrast, we utilize hypernetworks directly to accomplish image editing tasks. Moreover, our method can achieve both authentic attribute editing and cross-domain style editing simultaneously. To the best of our knowledge, this is the first occurrence of such an approach in the field of image editing.

Method

The overall structure of our model is illustrated in Figure 1, providing an overview of the image editing process. The following sections will comprehensively analyze each component in our approach.

The Design of Expressive Hypernetworks

Our approach is to reassign the generator's weights for image editing, which requires our hypernetworks to be expressive, allowing us to control the generator effectively. This control empowers us to edit images both authentically and cross-domain. The details of our hypernetworks are depicted in Figure 2. It takes the intermediate feature map $\hat{x} \in \mathbb{R}^{512\times16\times16}$ as input. Firstly, we extract the features of $\hat{x}$ and obtain $\hat{x}_i' \in \mathbb{R}^{512\times1\times1}$ through four down-sampling convolution operations. We then reshape $\hat{x}_i'$ as $\hat{x}_i'' \in \mathbb{R}^{512}$, and it undergoes a series of fully connected operations. In the fully connected operations, we first employ two distinct fully connected layers to expand $\hat{x}_i''$ dimensions, yielding two different vectors: $\sigma_1(\hat{x}_i'') \in \mathbb{R}^{k\times k\times c_i^{in}}$ and $\sigma_2(\hat{x}_i'') \in \mathbb{R}^{k\times k\times c_i^{out}}$. Here, σ_1 and σ_2 represent the FC1 and FC2, respectively. Where C_i^{in} and C_i^{out} represent the number of channels per convolution kernel and the total number of convolution kernels in the i^{th} layer of StyleGAN2's generator, respectively. Then, we reshape them and calculate the inner product to obtain the vector $\widehat{\Delta}_i \in \mathbb{R}^{k\times k\times C_i^{out}\times C_i^{in}}$, which is as follows:

$$\widehat{\Delta}_i = R\left(\sigma_1\left(\hat{x}_i''\right)\right) \otimes R\left(\sigma_2\left(\hat{x}_i''\right)\right) \tag{1}$$

Where R denotes the reshape operation, and $\otimes$ represents multidimensional matrix multiplication. To expand the representation space of $\hat{\Delta}_i$, we conduct two consecutive fully connected (FC) operations on $\hat{\Delta}_i$, followed by reshaping, resulting in the weight factors $\Delta_i \in \mathbb{R}^{C_i^{out}\times C_i^{in}\times k\times k}$. We denote the weights of the i^{th} layer in StyleGAN2 as $\theta_i = \{\theta_i^{j,k} \mid 0 \le j \le C_i^{out}, 0 \le k \le C_i^{in}\}$, where j represents the j^{th} convolutional kernel, and k denotes the k^{th} channel of the convolutional kernel. To achieve image editing, we reassign the weights of the generator based on equation 2. At this point, we obtain the final generated image after editing $y = G(\hat{\theta}, w_{init})$, where w_{init} is the latent vector of the original image x.

$$\hat{\theta}_i^{j,k} = \theta_i^{j,k} + \Delta_i^{j,k} \cdot \theta_i^{j,k} \tag{2}$$

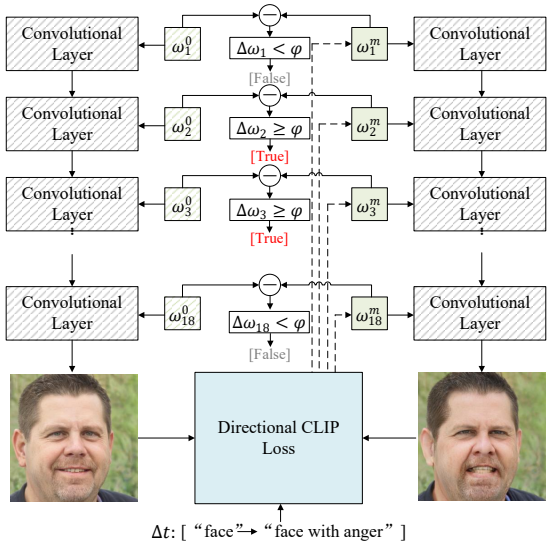

Figure 3: Overview of the adaptive layer selector. We utilize random latent codes as optimization objects and employ directional CLIP loss for a few iterations to dynamically select layers with significant differences between target and source codes. The grid part indicates parameter freezing, while the solid-colored part indicates optimization training.

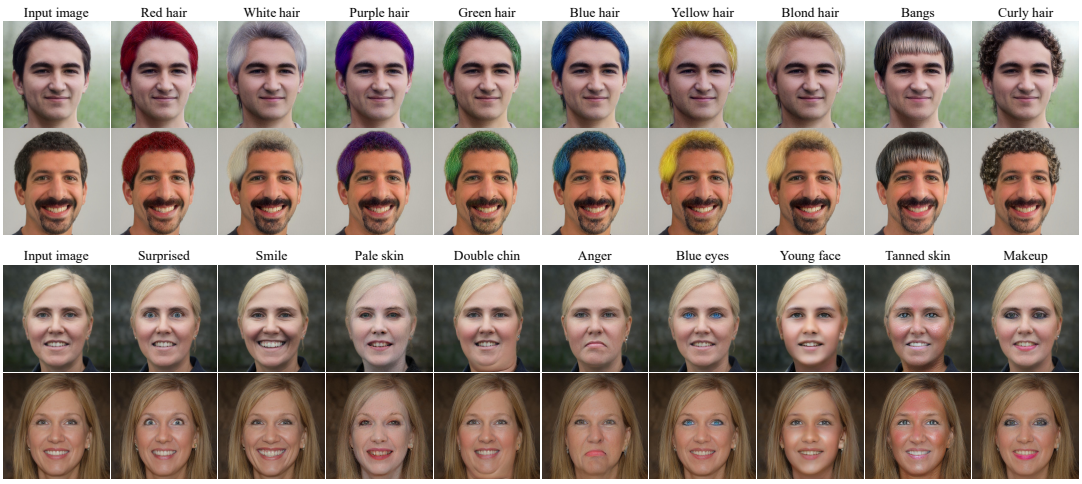

Figure 4: Results of image editing on FFHQ dataset by using our method. The target attributes are located above the images.

Directional CLIP Guidance

In the real world, pairwise image datasets where specific attributes change are scarce. It poses a challenge to supervise the alignment of generated images with target images efficiently. To tackle this issue, we leverage the CLIP model's potent cross-modal semantic alignment capability to facilitate the transformation of original images into target images in a self-supervised learning manner. In other words, the training of the entire model can be accomplished using only the original image and the target text prompt. In the existing work, (Wei et al. 2022) utilizes CLIP to directly align the generated image with the text conditions, as depicted in equation 3, where E_i represents the image encoder of CLIP, E_t represents the text encoder of CLIP, $cos(\cdot,\cdot)$ represents the cosine similarity, and y is the generated image.

$$\mathcal{L}_{CLIP}^{globe} = 1 - \cos\left(E_i(y), E_t(Text)\right) \tag{3}$$

Since specific attribute changes typically involve localized modifications, direct semantic alignment may lead to global feature alterations, making it challenging for the network to converge quickly. To address this issue, we draw inspiration from the approaches in (Kwon and Ye 2022; Lyu et al. 2023) and introduce the directional CLIP loss to align the text condition with the image. The process of semantic alignment is illustrated in equation 4, where T_y and T_x represent the target and source text, respectively (e.g., ["face with smile", "face"]). At this stage, we only need to ascertain the CLIP feature direction between the original and edited images. As the model continues to train, the generated image solely changes in this direction, ensuring that other local regions remain unaffected.

$$\mathcal{L}_{CLIP}^{direction} = 1 - \cos(E_i(y) - E_i(x), E_t(T_y) - E_t(T_x)) \tag{4}$$

Furthermore, in the feature extraction stage of the input image, we introduce a fusion modulation module to integrate the text conditions into the input feature map of the hypernetworks. By modifying the numerical characteristics of the intermediate layer feature map $\bar{x}$, we achieve indirect control over hypernetworks, allowing it to generate various weight factors and enabling a single model to produce diverse editing effects. In the fusion modulation module, to maintain consistency between the text condition and the generated image, we also incorporate the CLIP feature direction Δt between the texts as the conditional embedding. The embedding process is as follows:

$$\hat{x} = \bar{x} \cdot \alpha(\Delta t) + \beta(\Delta t), \text{where} \Delta t = E_t\left(T_y\right) - E_t\left(T_x\right) \tag{5}$$

Here, $\alpha(\cdot)$ and $\beta(\cdot)$ refer to the multi-layer perceptrons (MLPs), and $\bar{x}$ denotes the feature map obtained from the input image x through a series of operations such as convolution, batch normalization, and ResNet34 (He et al. 2016).

Adaptive Layer Selector

Recently, (Xia et al. 2021) demonstrated that different layers of the generator in StyleGAN2 control different attributes. Inspired by their findings, we question whether weight factors for all layers play a role in editing a single attribute. To explore this, we monitored how the weight factors generated by the hypernetworks changed during training. Figure 9 indicates that only a few layers undergo significant changes in the weight factors. Thus, we propose the adaptive layer selector, which allows hypernetworks to determine the layers requiring output weight factors autonomously. This technique alleviates the model's parameter count and maximizes the hypernetworks' efficiency when a single model only needs to implement a single attribute edit operation.

The schematic diagram of the adaptive layer selector is illustrated in Figure 3. Before training the model, we identify layers that exhibit significant variations in latent codes through latent optimization (Patashnik et al. 2021). These layers are then used as the selected layers to output weight factors. We found that the process of optimization can be completed with a few iterations, taking approximately less than 5 seconds. We first sample random noise $Z \sim N(0,1)$ to generate the initial latent codes ω_i^0. Then we optimize the

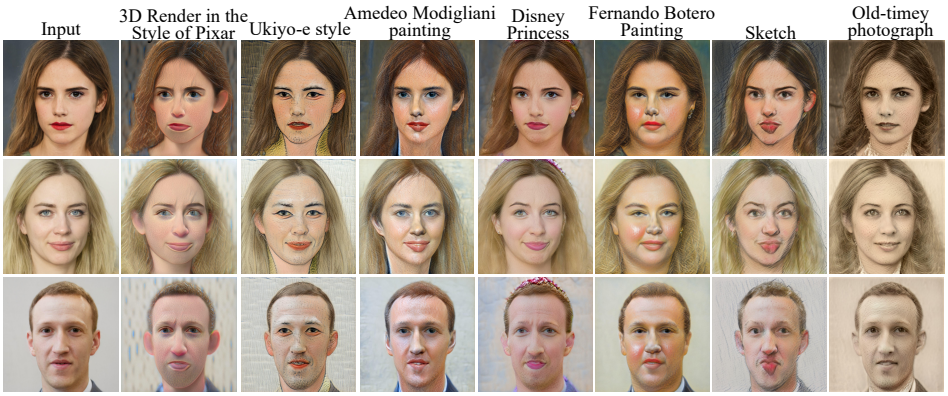

Figure 5: Results of cross-domain style editing by using our method. The target styles are located above the images.

initial latent codes ω_i^0 for m iterations using the directional CLIP loss to obtain ω_i^m. Now, we can obtain the difference set between the latent codes before and after optimization as $\Delta\omega_i = \left|\omega_i^m - \omega_i^0\right|$. To adaptively select the suitable layer, we establish an adaptive threshold as follows:

$$\varphi = \frac{\sum_{i=1}^{n} \Delta\omega_i}{n} + \lambda_{std} \sqrt{\frac{\sum_{j=1}^{n} \left(\Delta\omega_j - \frac{\sum_{i=1}^{n} \Delta\omega_i}{n}\right)^2}{n}} \quad (6)$$

Where λ_{std} represents the trade-off coefficient, and $n = 17$. If $\Delta\omega_i \geq \varphi$, then the i^{th} layer is considered the layer requiring output weight factors, and its index is added to the sequence L. Otherwise, no further action is taken. Finally, we obtain the sequence $L = \{i_1, i_2, \ldots, i_n\}$.

Loss Functions

Our objective is to modify specific target regions of the images while preserving the non-target regions unchanged. To achieve this, we follow the previous approach (Alaluf et al. 2022) and use a similarity loss in our work. The calculation process is as follows:

$$\mathcal{L}_{Sim} = 1 - cos(R_{Sim}(G(\theta + \Delta \cdot \theta, w_{init})), R_{Sim}(G(\theta, w_{init}))) \quad (7)$$

R_{Sim} represents the pre-trained ArcFace network (Deng et al. 2019) when the initial images belong to the facial domain, and the pre-trained MoCo model (He et al. 2020) when initial images belong to the non-facial domain. Additionally, we minimize the variations in global regions through L2 loss. The calculation process is as follows:

$$\mathcal{L}_2 = \|G(\theta + \Delta \cdot \theta, w_{init}) - G(\theta, w_{init})\|_2 \quad (8)$$

Combined with our goal of text-driven image editing, we define our comprehensive loss function as follows:

$$\mathcal{L} = \lambda_{clip}\mathcal{L}_{CLIP}^{direction} + \lambda_{norm}\mathcal{L}_2 + \lambda_{Sim}\mathcal{L}_{Sim} \quad (9)$$

Where λ_{clip} and λ_{norm} are both set to 1. And λ_{Sim} can take the values of either 0.1 or 0.5, depending on whether R_{Sim} corresponds to the ArcFace or MoCo networks. Notably, the ArcFace and MoCo networks cannot be simultaneously used during the model training process.

Experiments

Implementation Details

To validate the effectiveness of our approach, we conducted extensive experiments on diverse and challenging datasets. For the face domain, we utilized the FFHQ (Karras, Laine, and Aila 2018) dataset with 70,000 images as the training set and the Celeba-HQ (Karras et al. 2018) dataset with 28,000 images as the test set. Additionally, in the supplementary material, we provided image editing results on the Cat, Horse, Car, and Church datasets of LSUN (Yu et al. 2015), as well as the Cat, Dog, and Wild datasets of AFHQ (Choi et al. 2020). It is worth noting that all real images were inverted using the e4e encoder (Tov et al. 2021) to obtain the latent codes, and all generated images were produced using the pre-trained StyleGAN2 generator. Our training was conducted on a 4090 GPU, with a batch size of 4, and the Ranger optimizer, using a learning rate of 0.001.

Qualitative Evaluation

Results of authentic images editing. To validate the effectiveness of our method in utilizing various text conditions for image editing, we present the results of using text to control 18 different attributes in the image, as shown in Figure 4. The results demonstrate that our method effectively controls specific attributes while preserving irrelevant ones. More editing results from various datasets are shown in the supplementary material.

Results of cross-domain images editing. Figure 5 shows the results of our method editing real images to target images of different domains. The target domain never appears in the training process, which indicates that our method has good domain generalization ability. More editing results of cross-domain images editing are shown in the supplementary material.

Comparisons with the SOTA. In Figure 6, We compare our method with three current state-of-the-art ap-

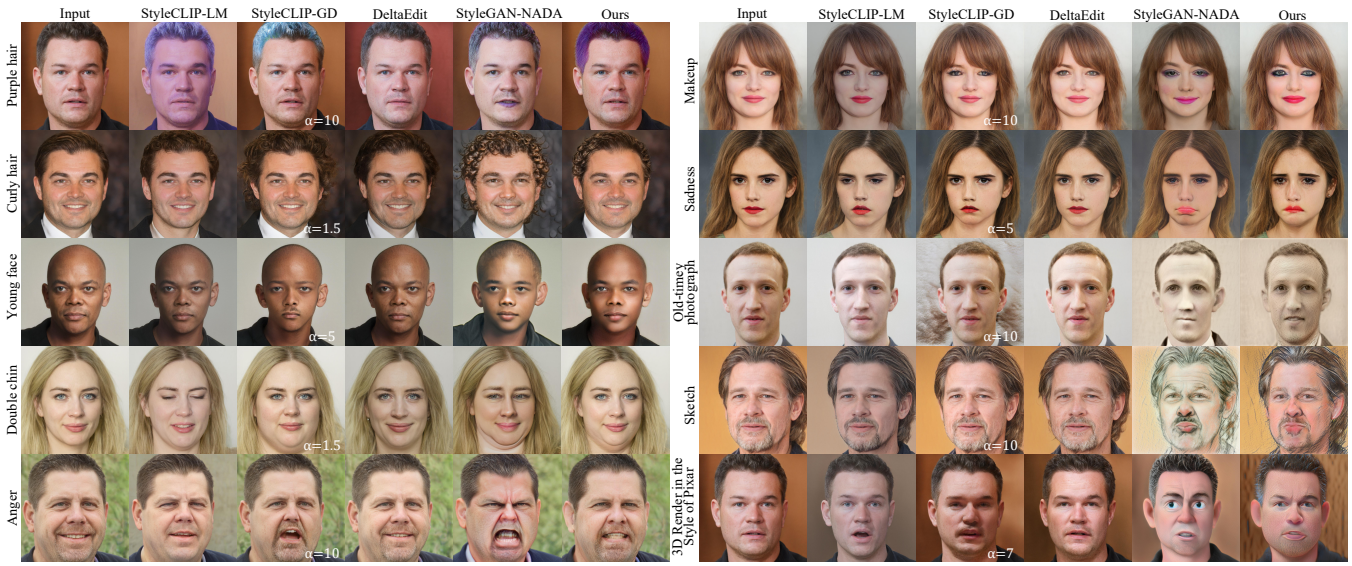

Figure 6: The results of comparing our method with DeltaEdit (Lyu et al. 2023), StyleCLIP-LM (Patashnik et al. 2021), StyleCLIP-GD (Patashnik et al. 2021), and StyleGAN-NADA (Gal et al. 2022) for image editing.

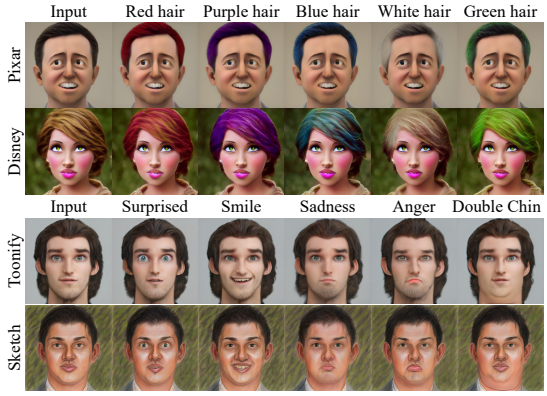

Figure 7: The weight factors produced by HyperEditor trained on the FFHQ dataset are applied to other domain generators (e.g., StyleGAN-NADA (Gal et al. 2022)).

proaches, namely DeltaEdit (Lyu et al. 2023), StyleGAN-NADA (Gal et al. 2022), and StyleCLIP (Patashnik et al. 2021). Among these methods, DeltaEdit and StyleCLIP utilize the manipulation of latent codes for image editing. In contrast, StyleGAN-NADA achieves style transfer by fine-tuning the generator's weights through retraining. In StyleCLIP, due to the extensive time consumption caused by optimization-based methods, here we only consider two methods based on latent mapper and global directions, respectively named StyleCLIP-LM and StyleCLIP-GD. Compared with DeltaEdit, where the editing results remain unchanged from the input image for attributes like "Purple hair" and "Young face", our method excels at achieving accurate modifications of specific image attributes. Compared with StyleCLIP-LM, our method not only edits more accurately, but also protects non-relevant regions better. Compared with StyleCLIP-GD, our method achieves better image editing results without parameter adjustment. Furthermore, compared to the approaches that manipulate the latent codes, our method can accomplish cross-domain image editing, while they cannot. Nevertheless, while StyleGAN-NADA excels in style transfer, our method outperforms it regarding the controllability of detailed attribute editing and the preservation of facial identity. More qualitative comparison results are shown in the supplementary material.

Weight factors' transferability. In this section, we apply the weight factors trained on the FFHQ dataset to generators in various domains. The edited images are depicted in Figure 7. The results demonstrate that the generated weight factors can be effectively transferred to generators in other domains, enabling authentic facial attribute editing without compromising the target style. More results are shown in the supplementary material.

Quantitative Evaluation

In Table 1, we provide the objective measures, including FID, PSNR, SSIM, LPIPS, IDS (identity similarity), and CS (CLIP score). All the results are the average values obtained by calculating the images before and after changing the ten attributes. Compared with state-of-the-art methods, our approach achieves the highest CLIP score (**27.35**), indicating that our results are more consistent with the target condition, confirming that HyperEditor can conduct more authentic image editing operations. Furthermore, in addition to achieving authentic editing, our method excels at preserving the image regions that are irrelevant to the editing target (as evidenced by Table 1, where we achieve the best results in the first four columns). Moreover, we calculated the FID values for the variations of "smile" and "double chin" attributes, which are displayed in Table 1. The minimal FID values signify

Methods	PSNR↑	LPIPS↓	SSIM↑	IDS↑	FID-0↓	FID-1↓	CS↑	Nop↓
StyleCLIP-GD $\alpha = 5$	24.96	0.28	0.79	0.72	11.73	44.05	24.82	-
StyleCLIP-GD $\alpha = 10$	20.51	0.33	0.76	0.64	17.08	201.95	25.28	-
StyleCLIP-LM	20.57	0.23	0.81	0.84	8.49	14.58	26.35	33.52M
StyleGAN-NADA	18.75	0.27	0.74	0.58	56.20	59.54	26.69	-
DeltaEdit	23.31	0.23	0.82	0.81	10.04	8.6	22.89	82.76M
Ours	**25.33**	**0.22**	**0.85**	**0.85**	**6.19**	**7.19**	**27.35**	71.48M
Ours(Global-CLIP)	23.21	0.39	0.77	0.71	120.18	61.07	22.96	-
Ours(w/ adaptive layer selector)	24.87	0.18	0.87	0.83	8.84	7.5	26.37	15.66M

Table 1: Quantitative evaluation of edited face images. Where FID-0 and FID-1 are obtained by calculating 2000 images before and after editing the "smile" and "double chin" attributes, respectively, the other values are obtained by computing the mean of images before and after editing for the ten different facial attributes. Nop denotes the number of model parameters.

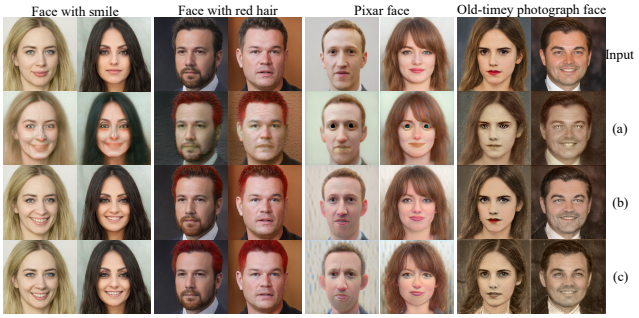

Figure 8: The results are presented in three cases: (a) without the adaptive layer selector and with Global-CLIP guidance, (b) with the adaptive layer selector and Directional-CLIP guidance, and (c) without the adaptive layer selector and with Directional-CLIP guidance.

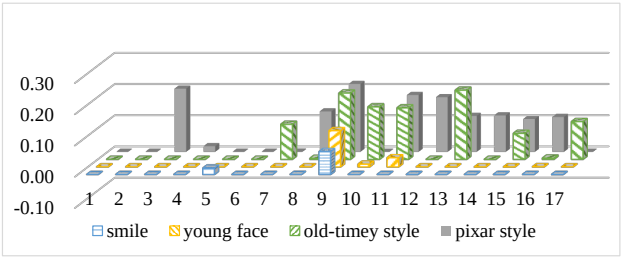

Figure 9: The variation in the average weight factors for each layer before and after editing specific attributes. The horizontal axis represents the layer index.

the closest distribution between the images produced by our method and the original images, and it also reflects the protection of non-edited regions by our approach.

Ablation Studies

Effectiveness of directional CLIP loss. The Global-CLIP guided text-driven image editing results are presented in Table 1 and Figure 8. The results indicate that the Global-CLIP guided method disrupts the global characteristics of the original image, leading to either significant differences between the generated image and the original image or blurriness. We attribute this to CLIP causing perturbations to the global feature semantics while guiding the local feature semantics to change. Additionally, (Gal et al. 2022) mentions the occurrence of adversarial interference during the image generation process guided by Global-CLIP. In contrast, our directional CLIP loss effectively protects the global feature semantics and provides more stable supervised training.

Rationality of adaptive layer selector. In Figure 9, we have documented the fluctuations in the average weight factors for each layer from the initiation of hypernetwork training until convergence. The results reveal that only some layers' weight factors undergo changes, confirming that it is not necessary to output weight factors for all layers during the image editing process. Table 1 and Figure 8 present results of image editing when we utilize the adaptive layer selector, at this point, $\lambda_{std} = 0.6$. The results demonstrate that using the adaptive layer selector can reduce the parameter count of hypernetworks by 80%, while still achieving equivalent results. Note that the adaptive layer selector is only suitable for a single model to edit a single attribute. If a single model is used to edit multiple attributes, the weight factors of all layers can be effectively used, so there is no need to reduce the number of layers that output weight factors. The selection of λ_{std} is illustrated in the supplementary material.

Conclusions

In this paper, we propose HyperEditor, an innovative framework that leverages hypernetworks to reassign weights of StyleGAN2's generator and utilizes CLIP for supervised training. As a result, compared with previous methods that achieve image editing by manipulating latent codes, our HyperEditor enables both authentic attribute editing and cross-domain style transfer. Compared to fine-tuning the generator by retraining, reassigning the generator's weights using hypernetworks offers more excellent controllability, enabling it to achieve finer precision in attribute editing and safeguarding the coherence of non-edited regions. Moreover, our fusion modulation module allows diverse editing operations within a single model, and the adaptive layer selector can reduce the model's complexity while editing a single attribute. Our innovative approach will open up the possibility to edit one thing to anything in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61871186 and 61771322.

References

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.

Choi, Y.; Uh, Y.; Yoo, J.; and Ha, J.-W. 2020. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8188–8197.

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

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

Gal, R.; Patashnik, O.; Maron, H.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4): 1–13.

Ha, D.; Dai, A. M.; and Le, Q. V. 2017. HyperNetworks. In *International Conference on Learning Representations*.

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

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, X.; Shen, L.; Patashnik, O.; Cohen-Or, D.; and Huang, H. 2022. Feat: Face editing with attention. *arXiv preprint arXiv:2202.02713*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*.

Karras, T.; Laine, S.; and Aila, T. 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4396–4405.

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.

Kocasari, U.; Dirik, A.; Tiftikci, M.; and Yanardag, P. 2022. StyleMC: multi-channel based fast text-guided image generation and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 895–904.

Kwon, G.; and Ye, J. C. 2022. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18062–18071.

Littwin, G.; and Wolf, L. 2019. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1824–1833.

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

Liu, Z.; Li, M.; Zhang, Y.; Wang, C.; Zhang, Q.; Wang, J.; and Nie, Y. 2023. Fine-Grained Face Swapping via Regional GAN Inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8578–8587.

Lyu, Y.; Lin, T.; Li, F.; He, D.; Dong, J.; and Tan, T. 2023. Deltaedit: Exploring text-free training for text-driven image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6894–6903.

Nirkin, Y.; Wolf, L.; and Hassner, T. 2021. Hyperseg: Patch-wise hypernetwork for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4061–4070.

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

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

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

Revanur, A.; Basu, D.; Agrawal, S.; Agarwal, D.; and Pai, D. 2023. CoralStyleCLIP: Co-Optimized Region and Layer Selection for Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12695–12704.

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

Saha, R.; Duke, B.; Shkurti, F.; Taylor, G. W.; and Aarabi, P. 2021. Loho: Latent optimization of hairstyles via orthogonalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1984–1993.

Shen, Y.; Gu, J.; Tang, X.; and Zhou, B. 2019. Interpreting the Latent Space of GANs for Semantic Face Editing.

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9240–9249.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33: 7462–7473.

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

von Oswald, J.; Henning, C.; Grewe, B. F.; and Sacramento, J. 2020. Continual learning with hypernetworks. In *International Conference on Learning Representations*.

Wei, T.; Chen, D.; Zhou, W.; Liao, J.; Tan, Z.; Yuan, L.; Zhang, W.; and Yu, N. 2022. Hairclip: Design your hair by text and reference image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18072–18081.

Xia, W.; Yang, Y.; Xue, J.-H.; and Wu, B. 2021. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2256–2265.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zeng, Y.; Lin, Z.; and Patel, V. M. 2022. Sketchedit: Mask-free local image manipulation with partial sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5951–5961.

Zhang, C.; Ren, M.; and Urtasun, R. 2019. Graph HyperNetworks for Neural Architecture Search. In *International Conference on Learning Representations*.

Zhu, Y.; Liu, H.; Song, Y.; Yuan, Z.; Han, X.; Yuan, C.; Chen, Q.; and Wang, J. 2022. One model to edit them all: Free-form text-driven image manipulation with semantic modulations. *Advances in Neural Information Processing Systems*, 35: 25146–25159.