Delving StyleGAN Inversion for Image Editing: A Foundation Latent Space Viewpoint

Hongyu Liu[1] Yibing Song[2*] Qifeng Chen[1*]

[1]Hong Kong University of Science and Technology [2]AI[3] Institute, Fudan University

hliudq@cse.ust.hk yibingsong.cv@gmail.com cqf@ust.hk

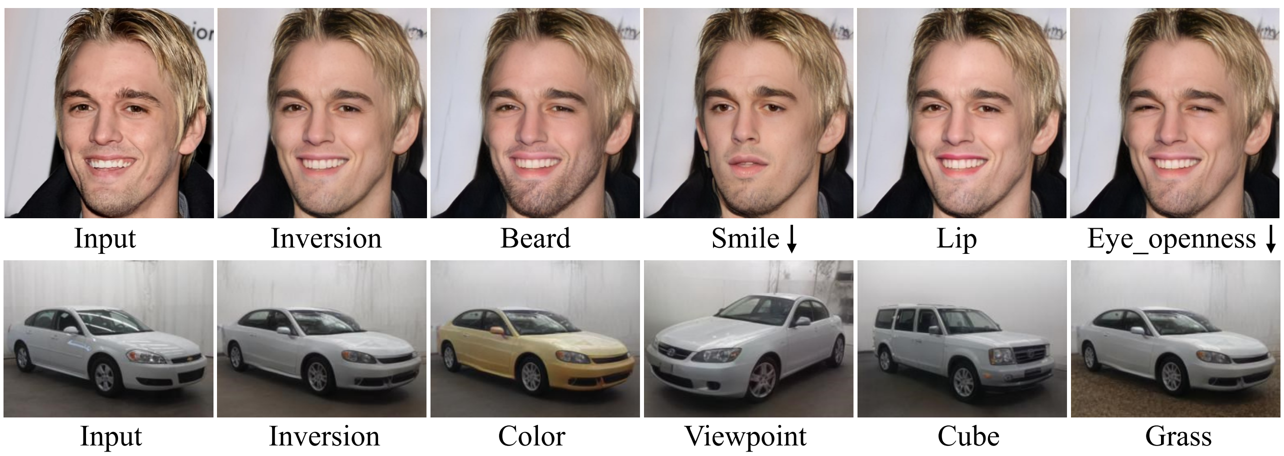

Figure 1. The inversion and editing results of our model in the real images. We show from the left to right of each row: an input image, inversion results, and our editing results. We edit images by modifying the attributes in the embedding space following [21, 54]. The ↓ means a decreased magnitude of the manipulation attribute.

Abstract

GAN inversion and editing via StyleGAN maps an input image into the embedding spaces ($\mathcal{W}$, $\mathcal{W}^+$, and $\mathcal{F}$) to simultaneously maintain image fidelity and meaningful manipulation. From latent space $\mathcal{W}$ to extended latent space $\mathcal{W}^+$ to feature space $\mathcal{F}$ in StyleGAN, the editability of GAN inversion decreases while its reconstruction quality increases. Recent GAN inversion methods typically explore $\mathcal{W}^+$ and $\mathcal{F}$ rather than $\mathcal{W}$ to improve reconstruction fidelity while maintaining editability. As $\mathcal{W}^+$ and $\mathcal{F}$ are derived from $\mathcal{W}$ that is essentially the foundation latent space of StyleGAN, these GAN inversion methods focusing on $\mathcal{W}^+$ and $\mathcal{F}$ spaces could be improved by stepping back to $\mathcal{W}$. In this work, we propose to first obtain the proper latent code in foundation latent space $\mathcal{W}$. We introduce contrastive learning to align $\mathcal{W}$ and the image space for proper latent code discovery. Then, we leverage a cross-attention encoder to transform the obtained latent code in $\mathcal{W}$ into $\mathcal{W}^+$ and $\mathcal{F}$, accordingly. Our experiments show that our exploration of the foundation latent space $\mathcal{W}$ improves the representation ability of latent codes in $\mathcal{W}^+$ and features in $\mathcal{F}$, which yields state-of-the-art reconstruction fidelity and editability results on the standard benchmarks. Project page: https://kumapowerliu.github.io/CLCAE.

1. Introduction

StyleGAN [29–31] achieves numerous successes in image generation. Its semantically disentangled latent space enables attribute-based image editing where image content is modified based on the semantic attributes. GAN inversion [62] projects an input image into the latent space, which benefits a series of real image editing methods [4, 36, 49, 65, 72]. The crucial part of GAN inversion is to find the inversion space to avoid distortion while enabling editability. Prevalent inversion spaces include the latent space $\mathcal{W}^+$ [1] and the feature space $\mathcal{F}$ [28]. $\mathcal{W}^+$ is shown to balance distortion and editability [56, 71]. It attracts many editing methods [1, 2, 5, 20, 25, 53] to map real images into this latent space. On the other hand, $\mathcal{F}$ contains spatial im-

*Y. Song and Q. Chen are the joint corresponding authors.

age representation and receives extensive studies from the image embedding [28, 48, 59, 63] or StyleGAN's parameters [6, 14] perspectives.

The latent space $\mathcal{W}^+$ and feature space $\mathcal{F}$ receive wide investigations. In contrast, Karras et al. [31] put into exploring $\mathcal{W}$ and the results are unsatisfying. This may be because that manipulation in $\mathcal{W}$ will easily bring content distortions during reconstruction [56], even though $\mathcal{W}$ is effective for editability. Nevertheless, we observe that $\mathcal{W}^+$ and $\mathcal{F}$ are indeed developed from $\mathcal{W}$, which is the foundation latent space in StyleGAN. In order to improve image editability while maintaining reconstruction fidelity (i.e., $\mathcal{W}^+$ and $\mathcal{F}$), exploring $\mathcal{W}$ is necessary. Our motivation is similar to the following quotation:

"You can't build a great building on a weak foundation. You must have a solid foundation if you're going to have a strong superstructure."

—Gordon B. Hinckley

In this paper, we propose a two-step design to improve the representation ability of the latent code in $\mathcal{W}^+$ and $\mathcal{F}$. First, we obtain the proper latent code in $\mathcal{W}$. Then, we use the latent code in $\mathcal{W}$ to guide the latent code in $\mathcal{W}^+$ and $\mathcal{F}$. In the first step, we propose a contrastive learning paradigm to align the $\mathcal{W}$ and image space. This paradigm is derived from CLIP [51] where we switch the text branch with $\mathcal{W}$. Specifically, we construct the paired data that consists of one image I and its latent code $w \in \mathcal{W}$ with pre-trained StyleGAN. During contrastive learning, we train two encoders to obtain two feature representations of I and w, respectively. These two features are aligned after the training process. During GAN inversion, we fix this contrastive learning module and regard it as a loss function. This loss function is set to make the one real image and its latent code w sufficiently close. This design improves existing studies [31] on $\mathcal{W}$ that their loss functions are set on the image space (i.e., similarity measurement between an input image and its reconstructed image) rather than the unified image and latent space. The supervision on the image space only does not enforce well alignment between the input image and its latent code in $\mathcal{W}$.

After discovering the proper latent code in $\mathcal{W}$, we leverage a cross-attention encoder to transform w into $w^+ \in \mathcal{W}^+$ and $f \in \mathcal{F}$. When computing w^+, we set w as the query and w^+ as the value and key. Then, we calculate the cross-attention map to reconstruct w^+. This cross-attention map enforces the value w^+ close to the query w, which enables the editability of w^+ to be similar to that of w. Besides, w^+ is effective in preserving the reconstruction ability. When computing f, we set the w as the value and key, while setting f as the query. So w will guide f for feature refinement. Finally, we use w^+ and f in StyleGAN to generate the reconstruction result.

We named our method CLCAE (i.e., StyleGAN inversion with **C**ontrastive **L**earning and **C**ross-**A**ttention **E**ncoder). We show that our CLCAE can achieve state-of-the-art performance in both reconstruction quality and editing capacity on benchmark datasets containing human portraits and cars. Fig. 1 shows some results. This indicates the robustness of our CLCAE. Our contributions are summarized as follows:

- We propose a novel contrastive learning approach to align the image space and foundation latent space $\mathcal{W}$ of StyleGAN. This alignment ensures that we can obtain proper latent code w during GAN inversion.
- We propose a cross-attention encoder to transform latent codes in $\mathcal{W}$ into $\mathcal{W}^+$ and $\mathcal{F}$. The representation of latent code in $\mathcal{W}^+$ and feature in $\mathcal{F}$ are improved to benefit reconstruction fidelity and editability.
- Experiments indicate that our CLCAE achieves state-of-the-art fidelity and editability results both qualitatively and quantitatively.

2. Related Work

2.1. GAN Inversion

GAN inversion [70] is the task to find a latent code in a latent space of pretrained-GAN's domain for the real image. As mentioned in the GAN inversion survey [62], the inversion methods can be divided into three groups: optimization-based, encoder-based, and hybrid. The optimization-based methods [1, 2, 7, 11, 11, 19, 64, 71] try to directly optimize the latent code or the parameters of GAN [53] to minimize the distance between the reconstruction image. The encoder-based methods [5, 10, 20, 25, 28, 33, 44, 50, 52, 56]learn a mapper to transfer the image to the latent code. The hybrid methods [69, 70] combine these two methods.

StyleGAN Inversion. Our work belongs to the StyleGAN inversion framework. Typically, there are three embedding spaces (i.e., $\mathcal{W}$ [30], $\mathcal{W}^+$ [1], and $\mathcal{F}$ [28]) and they are the trade-off design between the distortion and editability. The $\mathcal{W}$ is the foundation latent space of StyleGAN, several works [56, 71] have shown inverting the image into this space produces a high degree of editability but unsatisfied reconstruction quality. Differently, the $\mathcal{W}^+$ is developed from $\mathcal{W}$ to reduce distortions while suffering less editing flexibility. On the other hand, the $\mathcal{F}$ space consists of specific features in SyleGAN, and these features are generated by the latent input code of foundation latent space $\mathcal{W}$ in the StyleGAN training domain. The $\mathcal{F}$ space contains the highest reconstruction ability, but it suffers the worst editability. Different from these designs that directly explore $\mathcal{W}^+$ and $\mathcal{F}$, we step back to explore $\mathcal{W}$ and use it to guide $\mathcal{W}^+$ and $\mathcal{F}$ to improve fidelity and editability.

2.2. Latent Space Editing

Exploring latent space's semantic directions improves editing flexibility. Typically, there are two groups of methods to find meaningful semantic directions for latent space based editing: supervised and unsupervised methods. The supervised methods [3, 13, 17, 54] need attribute classifiers or labeled data for specific attributes. InterfaceGAN [54] use annotated images to train a binary Support Vector Machine [45] for each label and interprets the normal vectors of the obtained hyperplanes as manipulation direction. The unsupervised methods [21,55,58,65] do not need the labels. GanSpace [21] find directions use Principal Component Analysis (PCA). Moreover, some methods [24, 49, 60, 72] use the CLIP loss [51] to achieve amazing text guiding image manipulation. And some methods use the GAN-based pipeline to edit or inpaint the images [37–41].In this paper, we follow the [56] and use the InterfaceGAN and GanSpace to find the semantic direction and evaluate the manipulation performance.

2.3. Contrastive Learning

Contrastive learning [8, 15, 16, 18, 22, 47] has shown effective in self-supervised learning. When processing multi-modality data (i.e., text and images), CLIP [51] provides a novel paradigm to align text and image features via contrastive learning pretraining. This cross-modality feature alignment motivates generation methods [32, 49, 60, 61, 72] to edit images with text attributes. In this paper, we are inspired by the CLIP and align the foundation latent space $\mathcal{W}$ and the image space with contrastive learning. Then, we set the contrastive learning framework as a loss function to help us find the suitable latent code in $\mathcal{W}$ for the real image during GAN inversion.

3. Method

Fig. 3 shows an overview of the proposed method. Our CNN encoder is from pSp [52] that is the prevalent encoder in GAN inversion. Given an input image I, we obtain latent code w in foundation latent space $\mathcal{W} \in \mathbb{R}^{512}$. This space is aligned to the image space via contrastive learning. Then we set the latent code w as a query to obtain the latent code w^+ in $\mathcal{W}^+ \in \mathbb{R}^{N\times 512}$ space via $\mathcal{W}^+$ cross-attention block. The size of N is related to the size of the generated image (i.e., $N = 18$ when the size of the generated image is 1024×1024). Meanwhile, we select the top feature in the encoder as f in $\mathcal{F} \in \mathbb{R}^{H\times W\times C}$ space and use w to refine f with $\mathcal{F}$ cross-attention block. Finally, we send w^+ and f to the pretrained StyleGAN pipeline to produce the reconstruction results.

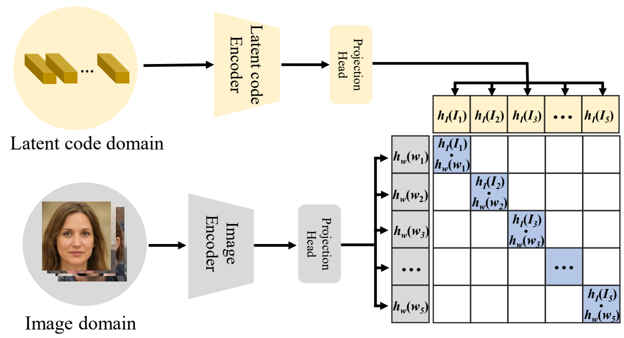

Figure 2. The process of contrastive learning pre-training. The encoders and projection heads extract the embedding of the image and latent code. Then we make the paired embeddings similar to align the image and latent code distribution. After alignment, we fix the parameters in the contrastive learning module to enable the latent code to fit the image during inversion.

3.1. Aligning Images and Latent Codes

We use contrastive learning from CLIP to align image I and its latent code w. After pre-training, we fix this module and use it as a loss function to measure the image and latent code similarity. This loss is set to train the CNN encoder in Fig. 3 as to align one image I and its latent code w.

The contrastive learning module is shown in Fig. 2. We synthesize 100K (I) and latent code(w) pairs with pre-trained StyleGAN. The I and w are fed into the module where there are feature extractors (i.e., CNN for I and transformer for w) and projection heads. Specifically, our minibatch contains S image and latent code pairs ($I \in \mathbb{R}^{256\times256\times3}$, $w \in \mathbb{R}^{512}$). We denote their embeddings after projection heads (i.e., hidden state) as $h_I(I) \in \mathbb{R}^{512}$ and $h_w(w) \in \mathbb{R}^{512}$, respectively. For the i-th pair from one minibatch (i.e., $i \in [1, 2, ..., S]$), its embeddings are $h_I(I_i)$ and $h_w(w_i)$. The contrastive loss [46, 68] can be written as

$$\mathcal{L}_i^{(I\to w)} = -\log \frac{\exp\left[\langle h_I(I_i), h_w(w_i)\rangle / t\right]}{\sum_{k=1}^{S} \exp\left[\langle h_I(I_i), h_w(w_k)\rangle / t\right]}, \quad (1)$$

$$\mathcal{L}_i^{(w\to I)} = -\log \frac{\exp\left[\langle h_w(w_i), h_I(I_i)\rangle / t\right]}{\sum_{k=1}^{S} \exp\left[\langle h_w(w_i), h_I(I_k)\rangle / t\right]}, \quad (2)$$

where $\langle\cdot\rangle$ denotes the cosine similarity, and $t \in \mathbb{R}^+$ is a learnable temperature parameter. The alignment loss in the contrastive learning module can be written as

$$\mathcal{L}_{\text{align}} = \frac{1}{S}\sum_{i=1}^{S}\left(\lambda\mathcal{L}_i^{(I\to w)} + (1-\lambda)\mathcal{L}_i^{(w\to I)}\right), \quad (3)$$

where $\lambda = 0.5$. We use the CNN in pSp [52] as the image encoder, and StyleTransformer [25] as the latent code

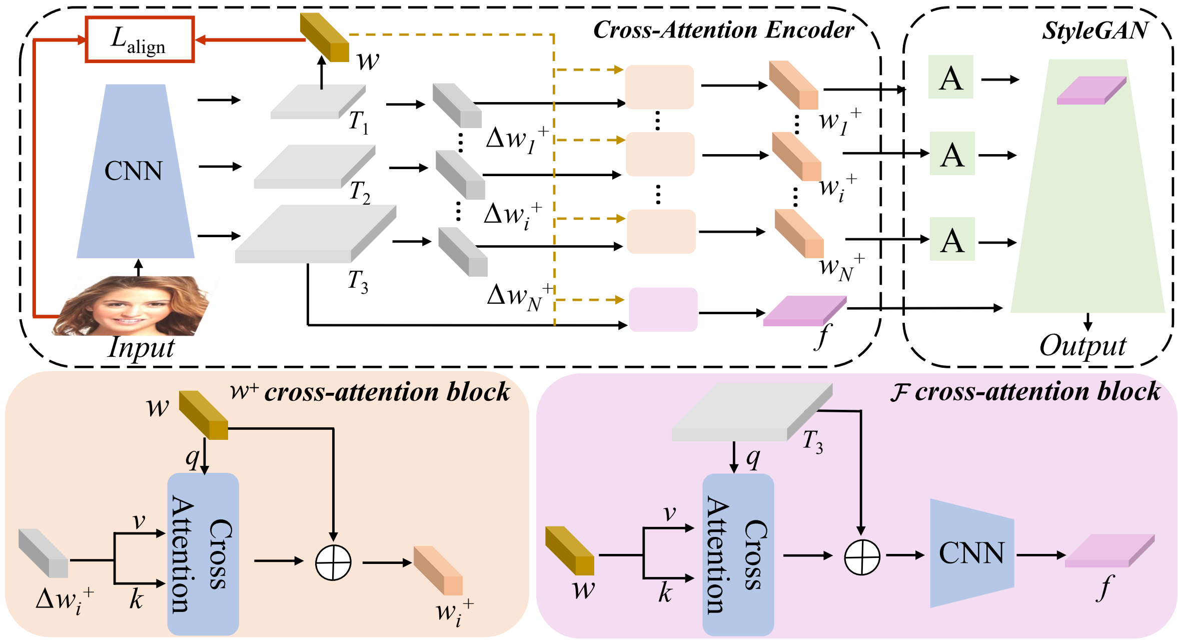

Figure 3. The pipeline of our method. With the input image, we first predict the latent code w with feature T_1. The w is constrained with the proposed $\mathcal{L}_{\text{align}}$. Then two cross-attention blocks take the refined w as a foundation to produce the latent code w^+ and feature f. Finally, we send the w^+ to StyleGAN via AdaIN [26] and replace the selected feature in StyleGAN with f to generate the output image.

encoder. Then in the GAN inversion process, we fix the parameters in the contrastive learning module and compute $\mathcal{L}_{align}$ to enable the latent code to fit the image. Aligning images to their latent codes directly via supervision $\mathcal{L}_{align}$ enforces our foundation latent space $\mathcal{W}^+$ close to the image space to avoid reconstruction distortions.

3.2. Cross-Attention Encoder

Once we have pre-trained the contrastive learning module, we make it frozen to provide the image and latent code matching loss. This loss function is utilized for training the CNN encoder in our CLCAE framework shown in Fig. 3. Our CNN encoder is a pyramid structure for hierarchical feature generations (i.e., T_1, T_2, T_3). We use T_1 to generate latent code w via a map2style block. Both the CNN encoder and the map2style block are from pSp [52]. After obtaining w, we can use I and w to produce an alignment loss via Eq. 3. This loss will further update the CNN encoder for image and latent code alignment. Also, we use w to discover w^+ and f with the cross-attention blocks.

3.2.1 $\mathcal{W}^+$ Cross-Attention Block

As shown in Fig. 3, we set the output of $\mathcal{W}^+$ cross-attention block as the residual of w to predict w^+. Specifically, we can get the coarse residual $\Delta w^+ \in \mathbb{R}^{N\times 512}$ with the CNN's features and map2style blocks first. Then we send each vector $\Delta w_i^+ \in \mathbb{R}^{512}$ in Δw^+ and $w \in \mathbb{R}^{512}$ to the $\mathcal{W}^+$ cross-attention block to predict the better Δw_i^+, where $i = 1, ..., N$. In the cross-attention block, we set the w as query(Q) and Δw_i^+ as value(V) and key(K) to calculate the attention map. This attention map can extract the potential relation between the w and Δw_i^+, and it can make the w^+ close to the w. Specifically, the Q, K, and V are all projected from Δw_i^+ and w with learnable projection heads, and we add the output of cross-attention with w to get final latent code w_i^+ in $\mathcal{W}^+$, the whole process can be written as

$$\begin{aligned}
&Q = wW_Q^{w^+}, K = \Delta w_i^+ W_K^{w^+}, V = \Delta w_i^+ W_V^{w^+},\\
&\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \qquad (4)\\
&w_i^+ = w + \text{Attention}(Q,K,V),
\end{aligned}$$

where $W_Q^{w^+}, W_K^{w^+}, W_V^{w^+} \in R^{512\times 512}$ and the feature dimension d is 512. We use the multi-head mechanism [57] in our cross-attention. The cross-attention can make the w^+ close to the w to preserve the great editability. Meanwhile, the reconstruction performance can still be preserved, since we get the refined w via the $\mathcal{L}_{\text{align}}$.

3.2.2 $\mathcal{F}$ Cross-Attention Block

The rich and correct spatial information can improve the representation ability of f as mentioned in [48]. We use the $T_3 \in \mathbb{R}^{64\times 64\times 512}$ as our basic feature to predict f as shown in Fig. 3, since the T_3 has the richest spatial information in the pyramid CNN. Then we calculate cross attention between the w and T_3 and output a residual to refine the T_3. In contrast to the W^+ cross-attention block, we set the w as value(V) and key(K) and T_3 as query(Q), this is because we want to explore the spatial information of w to support the T_3. Finally, we use a CNN to reduce the spatial size of the cross-attention block's output to get the final prediction f, the shape of f matches the feature of the selected convolution layer in $\mathcal{F}$ space. We choose the 5th convolution layer following the FS [63]. The whole process can be written as:

$$\begin{aligned} &Q = T_3 W_Q^f, K = w W_K^f, V = w W_V^f, \\ &\mathrm{Attention}(Q,K,V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \\ &f = \mathrm{CNN}\left[\mathrm{Attention}(Q,K,V) + T_3\right], \end{aligned} \tag{5}$$

where W_Q^f, W_K^f, $W_V^f \in R^{512\times512}$ and the feature dimension d is 512. Finally, we send the w^+ to the pretrained StyleGAN (G) via AdaIN [26] and replace the selected feature in G with f to get the final reconstruction result $G(w^+, f)$.

Image Editing. During the editing process, we need to get the modified $\hat{w}^+$ and $\hat{f}$. For the $\hat{w}^+$, we obtain it with the classic latent space editing methods [21, 54]. For the $\hat{f}$, we follow the FS [63] to generate the reconstruction result $G(w^+)$ and the edited image $G(\hat{w^+})$ respectively first. Then we extract the feature of the 5th convolution layer of $G(\hat{w^+})$ and $G(w^+)$ respectively. Finally, we calculate the difference between these two features and add it to the f to predict the $\hat{f}$. The whole process to get the $\hat{f}$ is:

$$\hat{f} = f + G^5(\hat{w}) - G^5(w), \tag{6}$$

where the $G^5(\tilde{w})$ and $G^5(w)$ is the feature of 5-th convolution layer. With the modified $\hat{w}^+$ and $\hat{f}$, we can get the editing results $G(\hat{w}^+, \hat{f})$.

3.3. Loss Functions

To train our encoder, we use the common ID and reconstruction losses to optimize the three reconstruction results $I_{rec}^1 = G(w)$, $I_{rec}^2 = G(w^+)$ and $I_{rec}^3 = G(w^+, f)$ simultaneously. Meanwhile, we use the feature regularization to make the f close to the original feature in G similar to the FS [63].

Reconstruction losses. We utilize the pixel-wise $\mathcal{L}_2$ loss and $\mathcal{L}_{\text{LPIPS}}$ [67] to measure the pixel-level and perceptual-level similarity between the input image and reconstruction image as

$$\mathcal{L}_{rec} = \sum_{i=1}^{3}\left(\lambda_{\text{LPIPS}}\,\mathcal{L}_{\text{LPIPS}}\left(I, I_{rec}^i\right) + \lambda_2\mathcal{L}_2\left(I, I_{rec}^i\right)\right), \tag{7}$$

where the $\mathcal{L}_{\text{LPIPS}}$ and $\mathcal{L}_2$ are are weights balancing each loss. We set the $\mathcal{L}_{\text{LPIPS}} = 0.2$ and $\mathcal{L}_2 = 1$ during training.

ID loss. We follow the e4e [56] to use the identity loss to preserve the identity of the reconstructed image as

$$\mathcal{L}_{id} = \sum_{i=1}^{3}\left(1 - \left\langle \mathrm{R}(I), \mathrm{R}(I_{rec}^i)\right\rangle\right). \tag{8}$$

For the human portrait dataset, the R is a pretrained ArcFace facial recognition network [27]. For the cars dataset, the R is a ResNet-50 [23] network trained with MOCOv2 [9].

Feature regularization. To edit the f with Eq. 6, we need to ensure f is similar to the original feature of G. So we adopt a regularization for the f as

$$\mathcal{L}_{f_{\text{reg}}} = \left\|f - G^5(w^+)\right\|_2^2. \tag{9}$$

Total losses. In addition to the above losses, we add the $\mathcal{L}_{\text{align}}$ to help us find the proper w. In summary, the total loss function is defined as:

$$\mathcal{L}_{total} = \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec}} + \lambda_{\text{ID}}\mathcal{L}_{\text{ID}} + \lambda_{f_{\text{reg}}}\mathcal{L}_{f_{\text{reg}}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}, \tag{10}$$

where λ_{rec}, λ_{ID}, $\lambda_{f_{\text{reg}}}$ and λ_{align} are the weights that adjust the contribution of each loss term. And we set the $\lambda_{\text{rec}} = 1$, $\lambda_{\text{ID}} = 0.1$, $\lambda_{f_{\text{reg}}=0.01}$ and $\lambda_{\text{align}} = 1$ respectively by default.

4. Experiments

In this section, we first illustrate our implementation details. Then we compare our method with existing methods qualitatively and quantitatively. Finally, an ablation study validates the effectiveness of our contributions. More results are provided in the supplementary files. We will release our implementations to the public.

4.1. Implementation Details

During the contrastive learning process, we follow the CLIP [51] and use the Adam optimizer [34] to train the image and latent code encoders. We synthesize the image-latent code pair dataset with the pre-trained StyleGAN2 in cars and human portrait domains. We set the batch size to 256 for training. During the StyleGAN inversion process, we train and evaluate our method on cars and human portrait datasets. For the human portrait, we use the FFHQ [30] dataset for training and the CelebA-HQ test set [43] for evaluation. For cars, we use the Stanford Cars [35] dataset

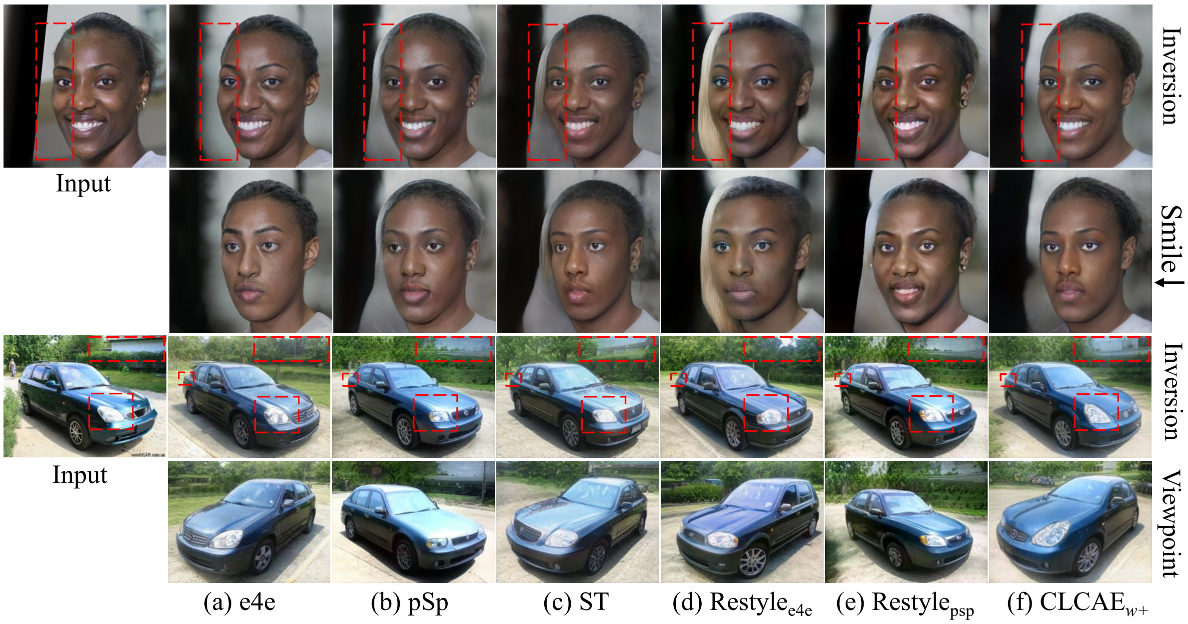

Figure 4. Visual comparison of inversion and editing between our method and the baseline methods (e4e [56], pSp [52], ST [25], restyle$_{e4e}$ [5] and restyle$_{pSp}$ [5]) in the $\mathcal{W}^+$ group. We produce CLCAE$_{w^+}$ $= G(w^+)$ to compare with them. Our method is more effective in producing manipulation attribute relevant and visually realistic results. ↓ means a reduction of the manipulation attribute.

for training and testing. We set the resolution of the input image as 256×256. We follow the pSp [52] and use the Ranger optimizer to train our encoder for GAN inversion, the Ranger optimizer is a combination of Rectified Adam [42] with the Lookahead technique [66]. We set the batch size to 32 during training. We use 8 Nvidia Telsa V100 GPUs to train our model.

4.2. Qualitative Evaluation

Our CLCAE improves the representation ability of the latent code in $\mathcal{W}^+$ and feature in $\mathcal{F}$ spaces. We evaluate qualitatively how our latent codes w^+ and f improve the output result. To clearly compare these two latent codes, we split the evaluation methods into two groups. The first group consists of methods only using latent code $w+$, we denote this group as 'group $\mathcal{W}^+$'. The second group consists of methods using both $w+$ and f, we denote this group as 'group $\mathcal{F}$'. When comparing to the group $\mathcal{W}^+$, we use our results CLCAE$_{w^+}$ computed via $G(w^+)$ for fair comparisons. When comparing to the group $\mathcal{F}$, we use our results computed via $G(w^+, f)$. During image editing, we use InterfaceGAN [54] and GanSpace [21] to find the semantic direction and manipulate the face and car images, respectively.

$\mathcal{W}^+$ space. Fig. 4 shows the visual results where our CLCAE$_{w^+}$ is compared to e4e [56], pSp [52], restyle$_{pSp}$ [5], restyle$_{e4e}$ [5] and StyleTransformer (ST) [25]. Both our CLCAE$_{w^+}$ and e4e show better inversion performance in the human portrait. This phenomenon is caused by the overfitting of those methods in (b)∼ (e), since the $\mathcal{W}^+$ space pays more attention to the quality of the reconstruction. The CLCAE$_{w^+}$ and e4e can produce w^+ close to the w, which improves the robustness of these two methods. Moreover, our CLCAE$_{w^+}$ is more capable of avoiding distortions while maintaining editability than other methods, including e4e (see the second row). This is because our w^+ is based on the solid w that does not damage the reconstruction performance of w^+. For the domain of cars, we observe that pSp and restyle$_{pSp}$ are limited to represent editing ability (see the (b) and (e) of the viewpoint row). On the other hand, e4e and ST are able to edit images, but their reconstruction performance are unsatisfying. In contrast to these methods, our CLCAE$_{w^+}$ maintains high fidelity and flexible editability at the same time.

$\mathcal{F}$ space. Fig. 5 shows our comparisons to PTI [53], Hyper [6], HFGI [59], and FS [63] in the $\mathcal{F}$ space. The results of PTI, Hyper, HFGI, and FS contain noticeable distortion in the face (e.g., the eyes in the red box regions in (a)∼ (d)) and the car (e.g., the background in (a)∼ (c) and the red box regions in car images). Although FS [63] reconstructs the background of the car image well, it loses editing flexibility (e.g., see (d) of 4 rows). This is because the FS method relies too much on $\mathcal{F}$ space, which limits the editability. In contrast, our results are in high fidelity as well as a wide range of editability with powerful f and w^+.

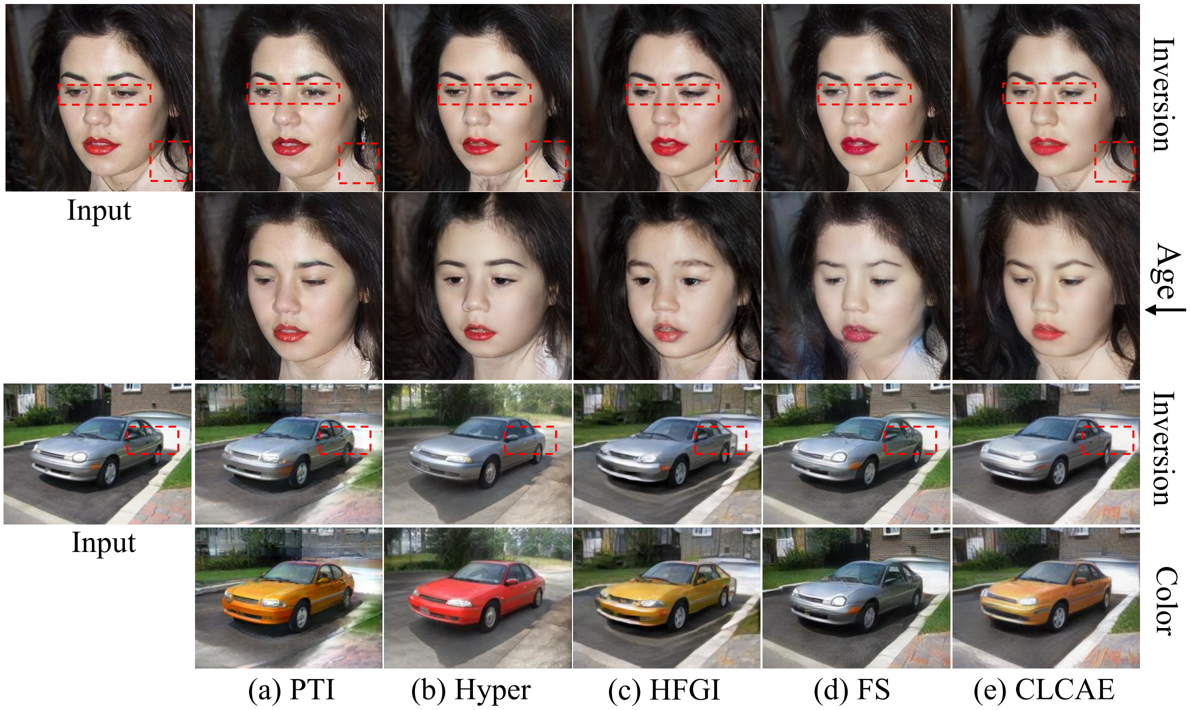

Figure 5. Visual comparison of inversion and editing between our method and the baseline methods (PTI [53], Hyper [6], HFGI [59], and FS [63]) in the $\mathcal{F}$ group. We produce CLCAE = $G(w^+, f)$ to compare with them. Our method not only generates high-fidelity reconstruction results but also retains the flexible manipulation ability. ↓ means a reduction of the manipulation attribute.

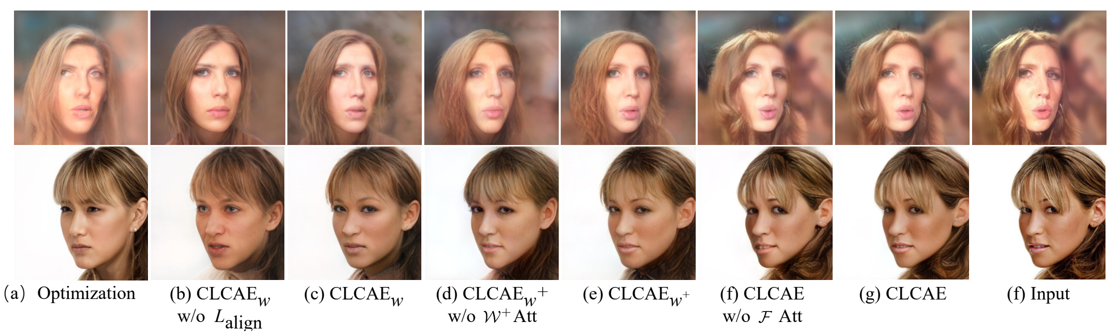

Figure 6. Visual results of ablation study. The (a) is an Optimization [31] method which inverts the image to the $\mathcal{W}$ space. The (b) and (c) are the results generated by w with and without $\mathcal{L}_{\text{align}}$ respectively. By comparing (a), (b), and (c), we can see that $\mathcal{L}_{\text{align}}$ can help our method produce better latent code w than optimization-based methods. (c) and (d) are the results generated by w^+ with and without $\mathcal{W}^+$ cross-attention block respectively. The (e) and (f) are the results generated by both w^+ and f with and without $\mathcal{F}$ cross-attention block, respectively. The performance gap between every two results can prove the effectiveness of w^+ and f cross-attention blocks.

4.3. Quantitative Evaluation

Inversion. We perform a quantitative comparison in the CelebA-HQ dataset to evaluate the inversion performance. We apply the commonly-used metric: PSNR, SSIM, LPIPS [67] and ID [27]. Table 1 shows these evaluation results. The PTI in $\mathcal{F}$ group and Restyle$_{pSp}$ in $\mathcal{W}^+$ group have better performance than our method in ID and LPIPS metric, respectively. But these two method takes a lot of time for the optimization operation or the iterative process. With the simple and effective cross-attention encoder and the proper foundation latent code, our method can achieve good performance in less time.

Editing. There is hardly a straight quantitative measurement to evaluate editing performance. We use the InterFaceGAN [54] to find the manipulation direction and edit the image, then we calculate the ID distance [27] between the original image and the manipulation one. For a fair comparison, during the ID distance evaluation, we use the

Table 1. Quantitative comparisons of state-of-the-art methods on the CelebA-HQ dataset. We conduct a user study to measure the editing performance. The number denotes the preference rate of our method against the competing methods. Chance is 50%. ↓ indicates lower is better while ↑ indicates higher is better.

Group		$\mathcal{W}^+$						$\mathcal{F}$				
Method		e4e [56]	pSp [52]	TS [25]	restyle$_{\text{e4e}}$ [5]	restyle$_{\text{pSp}}$ [5]	CLCAE$_{w^+}$	PTI [53]	Hyper [6]	HFGI [59]	FS [63]	CLCAE
Inversion	PSNR↑	19.08	20.39	20.50	19.45	21.20	**21.23**	23.49	22.09	22.13	24.08	**24.50**
	SSIM↑	0.53	0.56	0.57	0.54	0.57	**0.59**	0.65	0.61	0.62	0.67	**0.68**
	LPIPS↓	0.20	0.16	0.16	0.19	**0.13**	0.15	0.09	0.10	0.12	0.07	**0.06**
	ID↑	0.50	0.56	0.59	0.50	0.65	**0.65**	**0.83**	0.74	0.68	0.75	0.79
	Time↓	0.029s	0.028s	0.026s	1.154s	1.150s	0.071s	355.323s	1.161s	0.036s	0.581s	0.080s
Editing	ID↑ (Smile)	0.44	0.52	0.53	0.47	**0.64**	0.62	0.57	0.62	0.54	0.66	**0.67**
	User Study↓	70%	60%	62%	84%	73%	-	74%	72%	60%	96%	-

"smile" manipulation direction and adopt the same editing degree for CLCAE and other baselines. Besides using the object metric to evaluate the editing ability, we conduct a user study on the manipulated results from compared methods. We randomly collected 45 images of faces and cars for 9 groups of comparison methods, each group has 5 images, and these images are edited with our method and a baseline method, respectively. 20 participants need to select the one edited image with higher fidelity and proper manipulation. The user study is shown in Table 1. The results indicate that most participants support our approach.

4.4. Ablation Study

Effect of contrastive learning. We compare the optimization method [31] to evaluate whether our method can predict the solid latent code in foundation $\mathcal{W}$ space. The optimization method (a) can invert the image to the $\mathcal{W}$ with a fitting process. The visual comparisons are shown in Fig. 6, CLCAE$_w$ in (c) is the reconstruction results generated with our latent code w. Our method outperforms the optimization method in the ability of reconstruction and identity preservation. This is because the proposed L_{align} can directly calculate the distance between the latent code w and the image, while the optimization method only measures the difference in the image domain. Meanwhile, we present the results generated by w without L_{align} in (b) to prove our contrastive learning validity further. The associated numerical results are shown in Table 2.

Effect of the $\mathcal{W}^+$ Cross-Attention. To validate the effectiveness of $\mathcal{W}^+$ cross-attention block, we remove it and use the coarse residual as w^+ directly to do a comparison experiment. As shown in Fig. 6, the experiment results in (d) have distortion (see the eyes regions of the first row and the hair regions of the second row). And the cross-attention block in (e) can improve performance. This is because the cross-attention block utilizes the solid latent code w to support our method to predict better w^+. The numerical analysis results are shown in Table 2.

Effect of the $\mathcal{F}$ Cross-Attention. We analyze the effect of $\mathcal{F}$ cross-attention block by comparing the results produced with it and without it. We can see the visual comparison in Fig. 6. The results in (f) show that our method has artifacts in the hair and eye regions of the face without the $\mathcal{F}$ cross-attention block. And our method with $\mathcal{F}$ cross-attention block demonstrates better detail (see the hair and eyes in (g)). This phenomenon can prove that the $\mathcal{F}$ cross-attention block can extract the valid information in w and refine the f, which also tells us the importance of a good foundation. The numerical evaluation in Table 2 also indicates that $\mathcal{F}$ cross-attention block improves the quality of reconstructed content.

Table 2. Quantitative ablation study on the CelebA-HQ dataset. ↓ indicates lower is better while ↑ indicates higher is better.

Method	Optimization [31]	CLCAE$_w$ w/o $\mathcal{L}_{\text{align}}$	CLACE$_w$	CLCAE$_{w^+}$ w/o $\mathcal{W}^+$ Att	CLCAE$_{w^+}$	CLCAE w/o $\mathcal{F}$ Att	CLCAE
PSNR↑	16.95	18.15	**19.36**	20.61	**21.23**	23.93	**24.50**
SSIM↑	0.53	0.52	**0.54**	0.57	**0.59**	0.66	**0.679**
LPIPS↓	0.23	0.26	**0.22**	0.20	**0.15**	0.10	**0.06**
ID↑	0.19	0.26	**0.50**	0.56	**0.65**	0.70	**0.79**
Time↓	193.50s	0.022s	0.022s	0.028s	0.071s	0.074s	0.080s

5. Conclusion and Future Work

we propose a novel GAN inversion method CLCAE that revisits the StyleGAN inversion and editing from the foundation space $\mathcal{W}$ viewpoint. CLCAE adopts a contrastive learning pre-training to align the image space and latent code space first. And we formulate the pre-training process as a loss function $\mathcal{L}_{\text{align}}$ to optimize latent code w in $\mathcal{W}$ space during inversion. Finally, CLCAE sets the w as the foundation to obtain the proper w^+ and f with proposed cross-attention blocks. Experiments on human portrait and car datasets prove that our method can simultaneously produce powerful w, w^+, and f. In the future, we will try to expand this contrastive pre-training process to other domains (e.g., Imagenet dataset [12]) and do some basic downstream tasks such as classification and segmentation. This attempt could bring a new perspective to contrastive learning.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, 2020.

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *ACM Trans. Graph.*, 2021.

[5] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021.

[6] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[7] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[10] Edo Collins, R. Bala, B. Price, and S. Süsstrunk. Editing in style: Uncovering the local semantics of gans. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 2018.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

[13] Emily Denton, Ben Hutchinson, Margaret Mitchell, and Timnit Gebru. Detecting bias with generative counterfactual face attribute augmentation. *arXiv preprint arXiv:1906.06439*, 2019.

[14] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[15] Chongjian Ge, Youwei Liang, Yibing Song, Jianbo Jiao, Jue Wang, and Ping Luo. Revitalizing cnn attention via transformers in self-supervised visual representation learning. In *Advances in Neural Information Processing Systems*, 2021.

[16] Chongjian Ge, Jiangliu Wang, Zhan Tong, Shoufa Chen, Yibing Song, and Ping Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *International Conference on Learning Representations*, 2021.

[17] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 2020.

[19] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[20] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020.

[21] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Xianxu Hou, Linlin Shen, Or Patashnik, Daniel Cohen-Or, and Hui Huang. Feat: Face editing with attention. *arXiv preprint arXiv:2202.02713*, 2022.

[25] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. *arXiv preprint arXiv:2203.07932*, 2022.

[26] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[27] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[28] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations, 2021.

[29] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[32] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[33] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing, 2021.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 2013.

[36] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[37] Hongyu Liu, Xintong Han, ChengBin Jin, Huawei Wei, Zhe Lin, Faqiang Wang, Haoye Dong, Yibing Song, Jia Xu, and Qifeng Chen. Human motionformer: Transferring human motions with vision transformers. *arXiv preprint arXiv:2302.11306*, 2023.

[38] Hongyu Liu, Bin Jiang, Yibing Song, Wei Huang, and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 725–741. Springer, 2020.

[39] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4170–4179, 2019.

[40] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9371–9381, 2021.

[41] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, Jing Liao, Bin Jiang, and Wei Liu. Deflocnet: Deep image editing via flexible low-level controls. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10765–10774, 2021.

[42] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[43] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2015.

[44] Junyu Luo, Yong Xu, Chenwei Tang, and Jiancheng Lv. Learning inverse mapping by autoencoder based generative adversarial nets. In *International Conference on Neural Information Processing*, 2017.

[45] William S Noble. What is a support vector machine? *Nature biotechnology*, 2006.

[46] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[47] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[48] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multilayer selection for gan inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[49] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[50] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing, 2016.

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[52] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[53] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021.

[54] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[55] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[56] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation, 2021.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

[58] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*, 2020.

[59] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[60] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. Hairclip: Design your hair by text and reference image. *arXiv preprint arXiv:2112.05142*, 2021.

[61] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[62] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey, 2021.

[63] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A style-based gan encoder for high fidelity reconstruction of images and videos. *European conference on computer vision*, 2022.

[64] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with deep generative models, 2017.

[65] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[66] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. *Advances in neural information processing systems*, 2019.

[67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[68] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

[69] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *arXiv preprint arXiv:2004.00049*, 2020.

[70] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*. Springer, 2016.

[71] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2020.

[72] Yiming Zhu, Hongyu Liu, Yibing Song, Xintong Han, Chun Yuan, Qifeng Chen, Jue Wang, et al. One model to edit them all: Free-form text-driven image manipulation with semantic modulations. *arXiv preprint arXiv:2210.07883*, 2022.