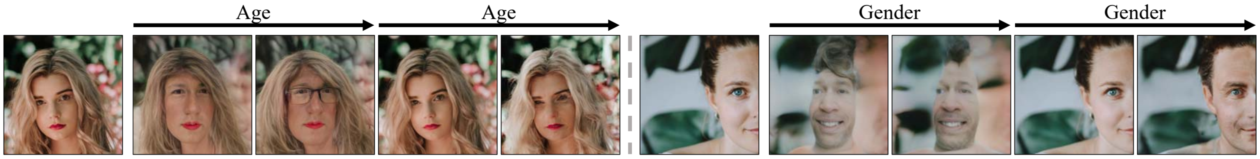
# GAN Inversion for Out-of-Range Images with Geometric Transformations

Kyoungkook Kang
POSTECH CSE
kkang831@postech.ac.kr

Seongtae Kim
POSTECH GSAI
seongtae0205@postech.ac.kr

Sunghyun Cho
POSTECH CSE & GSAI
s.cho@postech.ac.kr



(a) Input (b) PSP (c) Ours (d) Input (e) PSP (f) Ours

Figure 1: Qualitative comparison of reconstruction and semantic editing of different methods on in-the-wild images. (a) and (d) show input images, (b) and (e) show the results of PSP [22], and (c) and (f) show our results.

## Abstract

*For successful semantic editing of real images, it is critical for a GAN inversion method to find an in-domain latent code that aligns with the domain of a pre-trained GAN model. Unfortunately, such in-domain latent codes can be found only for in-range images that align with the training images of a GAN model. In this paper, we propose* BDInvert, *a novel GAN inversion approach to semantic editing of out-of-range images that are geometrically unaligned with the training images of a GAN model. To find a latent code that is semantically editable, BDInvert inverts an input out-of-range image into an alternative latent space than the original latent space. We also propose a regularized inversion method to find a solution that supports semantic editing in the alternative space. Our experiments show that BDInvert effectively supports semantic editing of out-of-range images with geometric transformations.*

## 1. Introduction

Generative adversarial networks (GANs) are generative models that can synthesize realistic-looking images [7]. Typically, GANs learn a mapping function from a random noise vector sampled from a pre-defined distribution to a realistic-looking image through the adversarial training of a generator and a discriminator. For the past several years, a significant progress has been made to improve the quality and diversity of synthesized images [21, 14, 15, 16, 5]. As a result, recent GAN models such as StyleGAN [15], StyleGAN2 [16], and BigGAN [5] can produce extremely high-quality images of high resolution.

Recently, it has been shown that rich semantic information is encoded in the intermediate features and the latent space of GANs, and furthermore, that images can be effectively edited in a semantically meaningful way by modifying features or latent code [21, 4, 26, 23, 10]. To enable such semantic editing for real images, GAN inversion has attracted much attention lately [3, 1, 31, 33]. GAN inversion maps a real image into the latent space of a pre-trained GAN model. Once an inverted latent code is obtained, the image can be semantically edited by modifying its latent code or intermediate features generated from the code.

For successful semantic editing of real images, it is critical to find an *in-domain* latent code that aligns with the domain of a pre-trained GAN model [31]. As shown in [31], there may exist more than one latent codes that can reconstruct a given input image, and some of them may be out of the domain. The semantic knowledge encoded in the latent space does not apply for such out-of-domain codes, thus semantic editing of such codes fails to produce proper results.

Unfortunately, such in-domain latent codes can be found only for a small fraction of real images that align with the training images of a pre-trained GAN model. For example, most GAN models use geometrically aligned face images as their training data for ease of training. As a result, images with a small amount of translation or other geometric transformations are out of their ranges, and the previous GAN inversion methods cannot find in-domain latent codes for such *out-of-range* images. This severely limits the applicability of semantic editing of real images using GAN inversion. Fig. 1 shows real-world examples. The input images in (a) and (d) are random images downloaded from internet. As they are out-of-range with different rotation, scaling and translation with respect to the training dataset (FFHQ [15]), directly applying a previous GAN inversion method [22] produces unacceptable results as shown in (b) and (e).

One solution would be to align a target image before GAN inversion, but accurate alignment of an image to the training data can be difficult or even impossible especially in the case of arbitrary natural images. For example, for the image in Fig. 1(d), a face alignment method [17] completely fails due to severe cropping.

In this paper, we propose a novel GAN inversion approach to semantic editing of *out-of-range* images, which is dubbed *Base-Detail Invert (BDInvert)*. BDInvert inverts a geometrically unaligned image with the training images for StyleGAN [15] and StyleGAN2 [16]. Specifically, BDInvert is designed to cover geometric transformations such as translation, rotation, and scaling, and supports various types of editing for out-of-range images that are not supported by previous approaches.

Our key idea is as follows. It is impossible to invert an out-of-range image to an in-domain latent code in the original latent space of a pre-trained GAN model. Instead, we propose to invert an image into another space that we refer to as the $\mathcal{F}/\mathcal{W}^+$, which consists of two subspaces $\mathcal{F}$ and $\mathcal{W}^+$. The base code space $\mathcal{F}$ encodes geometric transformations and also supports diverse local variations that enable more faithful reconstruction of an input image. On the other hand, the detail code space $\mathcal{W}^+$ is independent of geometric transformations and supports semantic manipulations.

To find a latent code in the $\mathcal{F}/\mathcal{W}^+$ space that faithfully reconstructs an input image, we adopt an optimization-based approach. However, naïve optimization of a reconstruction loss does not guarantee a latent code that supports semantic editing. To enable semantic editing, we also propose a regularization approach based on an encoder network. Fig. 1(c) and (f) show our reconstruction and editing results of real-world images. Thanks to our $\mathcal{F}/\mathcal{W}^+$ space and inversion approach, we can successfully reconstruct and edit the out-of-range real-world input images.

Our main contributions can be summarized as follows.

- We propose *BDInvert*, a novel GAN inversion approach to semantic editing of real images with geometric transformations that are not aligned with the training images of a pre-trained GAN model.
- BDInvert projects an image into an alternative latent space $\mathcal{F}/\mathcal{W}^+$ that supports more faithful reconstruction and semantic editing of out-of-range images with geometric transformations and diverse local variations.
- We propose a novel regularization method to find a proper solution in the $\mathcal{F}/\mathcal{W}^+$ space that supports semantic image editing.

## 2. Related Work

In order to embed real images into the latent space of GANs, various approaches have been proposed in two directions. One direction is to train an encoder using a data-driven approach [32, 9, 22]. The other direction is to initialize a latent vector randomly or to the output of a pre-trained encoder, and then to optimize it to reconstruct a target image [32, 28, 6, 8, 3]. However, inverting a real image remains a difficult problem because of the limited expressiveness of the latent space of GANs.

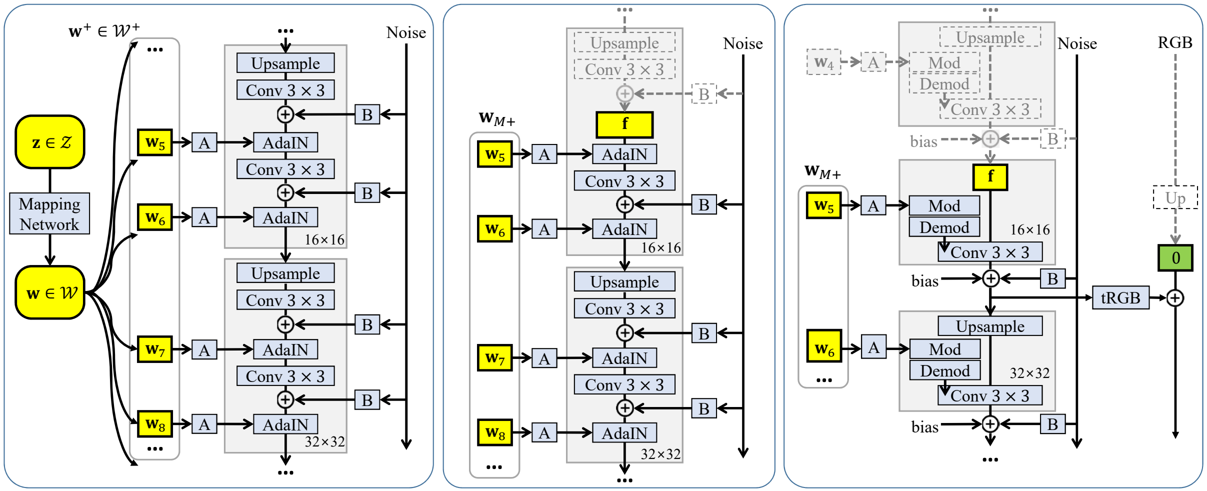
Recently, in order to enhance the inversion quality, several attempts to widen the latent space have been made [8, 20, 13]. Gu *et al.* [8] improved the reconstruction quality by mixing features from several latent codes. Pan *et al.* [20] fine-tune a generator on-the-fly for more faithful reconstruction. Huh *et al.* [13] find geometric transformation parameters to transform an image region to be more suitable for BigGAN [5] inversion. Meanwhile, Abdal *et al.* [1] showed high-quality embedding results for StyleGAN [15] using an extended latent space $\mathcal{W}^+$. Afterwards, many studies focusing on StyleGAN have been proposed [2, 33, 31, 16, 26]. Abdal *et al.* [2] and Karras *et al.* [16] optimize the noise channel for more accurate embedding. For successful image editing, embedding an image into GAN's domain is essential. To this end, Zhu *et al.* [31] train an encoder that projects an image into StyleGAN's domain, and optimize a latent code with the guidance of the encoder. Tewari *et al.* [26] introduced a hierarchical optimization that first embeds an image into the $\mathcal{W}$ space and then embeds it into the $\mathcal{W}^+$ space for better editing. Zhu *et al.* [33] proposed the $\mathcal{P} - norm^+$ space for in-domain inversion. However, most existing works cannot handle out-of-range images.

**Semantic editing** A widely used approach to semantic image editing using GAN is to modify a latent code along semantically meaningful directions. Härkönen *et al.* [10] identify semantic directions by applying the principal component analysis (PCA) on sampled latent codes. Shen *et al.* [23] use attribute classifiers to discover semantic directions. Shen and Zhou [24] proposed an unsupervised method that factorizes the weights of latent code transformation layers to find semantic directions that cause large changes to the output.

## 3. Latent Space $\mathcal{F}/\mathcal{W}^+$

In this section, we first review state-of-the-art GAN inversion approaches and discuss their limitations on out-of-range images. Then, we introduce an alternative latent space $\mathcal{F}/\mathcal{W}^+$ to overcome the limitations.

Our approach is based on StyleGAN and StyleGAN2 [15, 16], which produce high-quality synthesis results. Both GAN frameworks use a mapping network $f : \mathcal{Z} \rightarrow \mathcal{W}$ based on a multi-layer perceptron (MLP) that maps a latent code $\mathbf{z} \in \mathcal{Z}$ to an intermediate latent code $\mathbf{w} \in \mathcal{W}$ as shown in Fig. 2(a). Compared to the latent space $\mathcal{Z}$, the intermediate latent space $\mathcal{W}$ provides less entangled representations of different attributes so that different attributes can be more easily adjusted in the image generation process. Another noticeable feature of StyleGAN and StyleGAN2 is their multi-scale image synthesis approaches,



(a) StyleGAN and the latent spaces $\mathcal{Z}$, $\mathcal{W}$ and $\mathcal{W}^+$

(b) $\mathcal{F}/\mathcal{W}^+$ of StyleGAN

(c) $\mathcal{F}/\mathcal{W}^+$ of StyleGAN2

Figure 2: The network architectures of StyleGAN [15] and StyleGAN2 [16] and the layers corresponding to their latent spaces marked in yellow. The layers that are not used for the latent space $\mathcal{F}/\mathcal{W}^+$ are marked with gray dotted borders in (b) and (c). In StyleGAN2, the RGB image layer from the coarse scale is replaced with a tensor filled with zero as indicated by a green box.

which enable scale-wise disentanglement of different attributes. To control the generation process in a multi-scale manner, both StyleGAN and StyleGAN2 feed the intermediate latent code $\mathbf{w}$ to multiple layers of different scales of the generator. In addition, to enhance the diversity of synthesized images, both StyleGAN and StyleGAN2 utilize noise randomly sampled from a Gaussian distribution for each image generation.

While $\mathcal{W}$ is effective in generating diverse images with different attributes, it is still not sufficient for GAN inversion of a wide range of real images. To enhance the reconstruction accuracy, Abdal *et al.* [1] proposed an extended latent space $\mathcal{W}^+$. Each element $\mathbf{w}^+ \in \mathcal{W}^+$ is defined as $\mathbf{w}^+ = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$ where $\mathbf{w}_i$ is a latent code in $\mathcal{W}$ and $N$ is the number of layers in the generator that takes $\mathbf{w}$ as input (Fig. 2(a)). The subscript $i$ in $\mathbf{w}_i$ is the index of a layer that takes $\mathbf{w}$ such that $i = 1$ and $i = N$ indicate the first and last layers in the smallest and largest scales, respectively. With the extended latent space $\mathcal{W}^+$, different latent codes can be used for different layers, and consequently, a wider range of images can be reconstructed.

Later, Zhu *et al.* [31] showed that, for semantic image manipulation, it is essential to find an *in-domain* latent code instead of a latent code that precisely reconstructs an input image. They also showed that real images can be effectively inverted to an in-domain latent code in $\mathcal{W}^+$ with a domain-guided encoder and domain-regularized optimization.

Nonetheless, GAN inversion to the extended latent space $\mathcal{W}^+$ still fails to find an in-domain latent code for *out-of-range* images as discussed in Sec. 1. To overcome this limitation, we propose another latent space $\mathcal{F}/\mathcal{W}^+$. Each element $\mathbf{w}^*$ in $\mathcal{F}/\mathcal{W}^+$ is defined as $\mathbf{w}^* = (\mathbf{f}, \mathbf{w}_{M+})$ where $\mathbf{f}$ is a base code and $\mathbf{w}_{M+}$ is a detail code. $\mathbf{w}_{M+}$ is a set of latent codes for the fine scales of the generator, which is defined as $\mathbf{w}_{M+} = \{\mathbf{w}_M, \cdots, \mathbf{w}_N\}$. $\mathbf{f}$ is a coarse-scale feature map of the generator before the layer that takes $\mathbf{w}_M$. Specifically, for StyleGAN [15], we define $\mathbf{f}$ as the feature map right before the first adaptive instance normalization (AdaIN) layer [12] at a certain scale. For StyleGAN2 [16], we define $\mathbf{f}$ as the feature map after a pair of upsampling and convolution layers at a certain scale. Fig. 2(b) and (c) depict the latent space $\mathcal{F}/\mathcal{W}^+$ of StyleGAN and StyleGAN2, respectively. In our experiments, we test two different scales, $8 \times 8$ and $16 \times 16$, for $\mathbf{f}$.

In the case of StyleGAN2 [16], the generator needs a feature map corresponding to an RGB image upsampled from the previous scale (Fig. 2(c)). While we may include a small-scale feature map as a part of our latent space, we observed that the feature maps at the coarse scales have values close to zero and have little impact on image generation results. Thus, we simply set them to zero in our experiments as depicted by the green box in Fig. 2(c).

The $\mathcal{F}/\mathcal{W}^+$ space provides a couple of nice properties that enable semantic editing of out-of-range images. First, compared to $\{\mathbf{w}_1, \cdots, \mathbf{w}_{M-1}\}$, the base code $\mathbf{f}$ can represent a wider range of images including images with geometric transformations. For example, as $\mathbf{f}$ is a feature map of a convolutional neural network (CNN), we can simply shift $\mathbf{f}$ along the $x$- or $y$-axis to represent the feature map of a shifted image. Second, the detail code $\mathbf{w}_{M+}$ is invariant to translations of images. Specifically, in the case of StyleGAN [15], $\mathbf{w}_{M+}$ controls the parameters of the AdaIN [12] layers of the generator. Similarly, in the case of StyleGAN2 [16], $\mathbf{w}_{M+}$ controls the parameters of the demodulation layers. Both AdaIN and demodulation operations are global operations that are applied to CNN features

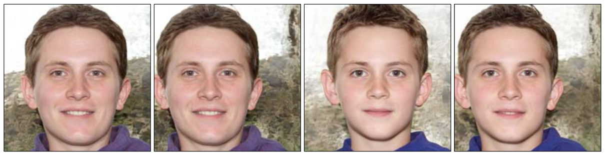

(a) An in-range synthesized image $G(\mathbf{f}, \mathbf{w}_{M+})$ (b) A synthesized image with a shifted $\mathbf{f}$ $G(T'(\mathbf{f}), \mathbf{w}_{M+})$ (c) semantic editing of (a) $G(\mathbf{f}, \mathbf{w}'_{M+})$ (d) semantic editing of (b) $G(T'(\mathbf{f}), \mathbf{w}'_{M+})$

Figure 3: Semantic editing in the $\mathcal{F}/\mathcal{W}^+$ space. (a) An in-range synthesized image generated from an in-domain latent code $(\mathbf{f}, \mathbf{w}_{M+})$. (b) By applying a geometric transform to $\mathbf{f}$, an out-of-range image can be obtained. Because $\mathbf{w}'_{M+}$ affects the image globally, an image editing operation used for an in-range image in (c) can be used for an out-of-range image in (d).

in a translation-invariant manner.

Thanks to the aforementioned properties, we can describe the relationship between an image $I$ and its transformed image $T(I)$ where $T$ is a geometric transformation operator as follows. Suppose that $I$ is generated from $\mathbf{w}^*$, i.e., $I = G(\mathbf{w}^*) = G(\mathbf{f}, \mathbf{w}_{M+})$ where $G$ is the generator of a pre-trained GAN model. Then, $T(I)$ can be expressed as:

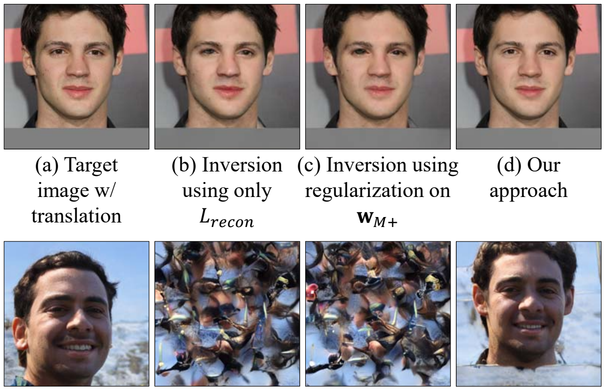
$$T(I) \approx G(T'(\mathbf{f}), \mathbf{w}_{M+}) \tag{1}$$

where $T'$ is a geometric transformation operator corresponding to $T$ whose scale is adjusted according to the relative scale of $\mathbf{f}$ to $I$. This relationship can be also used for semantic image manipulation of $T(I)$. As $T'(\mathbf{f})$ is a CNN feature map and $\mathbf{w}_{M+}$ is a set of parameters for global operations, for editing $T(I)$, we can manipulate $\mathbf{w}_{M+}$ in the same way for $I$ and achieve similar editing results.

Fig. 3 shows an example that illustrates the relationship in Eq. (1). In this example, we sample an in-domain latent code $(\mathbf{f}, \mathbf{w}_{M+})$ and generate an in-range image in Fig. 3(a) using StyleGAN2 [16]. Shifting $\mathbf{f}$, we can generate a shifted image of Fig. 3(a) as shown in Fig. 3(b). While they are not exactly the same due to the zero padding and noise component in StyleGAN2, they look almost identical proving the relationship in Eq. (1). Fig. 3(c) and (d) show the semantic editing results of (a) and (b) using the same manipulated latent code $\mathbf{w}'_{M+}$. The results show that we can effectively perform semantic editing for geometrically transformed images in the same way as for in-range images.

The discussion above shows that, as long as $(\mathbf{f}, \mathbf{w}_{M+})$ is in-domain, $(T'(\mathbf{f}), \mathbf{w}_{M+})$ for an arbitrary $T'$ also supports semantic image editing. Based on this, we define an extended domain of $\mathbf{w}^*$ as a set of geometrically transformed latent codes $(T'(\mathbf{f}), \mathbf{w}_{M+})$ of in-domain latent codes $(\mathbf{f}, \mathbf{w}_{M+})$ for arbitrary transformations $T'$.

While the discussion above discusses only geometric transformations, we note that our latent space $\mathcal{F}/\mathcal{W}^+$ supports not only geometric transformations but also diverse local variations as the base code $\mathbf{f}$ supports locally different information. This leads to more faithful reconstruction even for images without geometric transformations as will be shown in Sec. 5. We also note that the latent space $\mathcal{F}/\mathcal{W}^+$ does not support semantic editing that require coarse-scale $\mathbf{w}_i$'s such that $i < M$. However, our experiments show that it still supports various types of semantic editing as we define $\mathbf{f}$ as a very coarse-scale feature map.


(a) Target image w/ translation (b) Inversion using only $L_{recon}$ (c) Inversion using regularization on $\mathbf{w}_{M+}$ (d) Our approach

(e) Reference image for style mixing (f) Style mixing result of (b) (g) Style mixing result of (c) (h) Style mixing result of (d)

Figure 4: Inversion to $\mathcal{F}/\mathcal{W}^+$ with different combinations of the loss terms. The results of the style mixing [15] operation in (f), (g) and (h) are obtained by replacing their $\mathbf{w}_{M+}$'s by $\mathbf{w}_{M+}$ from the reference image in (e).

## 4. Regularized Inversion to $\mathcal{F}/\mathcal{W}^+$

For inversion of an image, we adopt the optimization-based approach since it generally achieves higher reconstruction quality compared to the encoder-based approach [6, 8, 31]. In this section, we introduce our optimization approach both for StyleGAN and StyleGAN2 [15, 16].

### 4.1. Reconstruction Loss

Given an input image $I$, to find a latent code $\mathbf{w}^*$ that reconstructs $I$, we optimize an objective function with a reconstruction loss $L_{recon}$, which is defined as:

$$L_{recon}(\mathbf{w}^*) = L_{MSE}(\mathbf{w}^*) + \omega_{per} L_{per}(\mathbf{w}^*) \tag{2}$$

where $L_{MSE}$ and $L_{per}$ are mean-squared-error (MSE) and perceptual losses, respectively. $\omega_{per}$ is a weight for $L_{per}$. $L_{MSE}$ is defined as $L_{MSE}(\mathbf{w}^*) = \|I - G(\mathbf{w}^*)\|^2$ where $G$ is the generator of a pre-trained StyleGAN model. $L_{per}$ is defined as $L_{per}(\mathbf{w}^*) = \|F(I) - F(G(\mathbf{w}^*))\|^2$, where $F$ is a LPIPS network to compute the perceptual distance [30].

By optimizing Eq. (2), e.g., using the Adam optimizer [18], we can obtain latent codes that produce high-quality reconstruction results even for out-of-range images thanks to the high expressive power of the $\mathcal{F}/\mathcal{W}^+$ space. However, such latent codes do not support semantic image editing as they are out-of-domain. Fig. 4 shows an example using StyleGAN2 [16]. Fig. 4(a) shows a target image, which is out-of-range due to translation. Fig. 4(e) is a reference image for style mixing, which is a semantic image editing operation [15]. Optimizing the reconstruction loss in Eq. (2), we can obtain a latent code that accurately reconstructs the target image as shown in Fig. 4(b). However, the estimated latent code is out-of-domain, so it fails to produce an appropriate style mixing result as shown in Fig. 4(f).

To enable semantic editing of out-of-range images, both $\mathbf{f}$ and $\mathbf{w}_{M+}$ must be in proper domains. To guide our optimization process to a solution in a proper domain, we adopt regularization both on $\mathbf{f}$ and $\mathbf{w}_{M+}$. The following subsections discuss our regularization schemes one by one.

### 4.2. Regularization on Detail Code $\mathbf{w}_{M+}$

To promote in-domain $\mathbf{w}_{M+}$, we adopt the $\mathcal{P}-norm^{+}$ space-based regularization scheme proposed by Zhu *et al*. [33]. Specifically, at each iteration of the iterative optimization of our objective function, we transform the current estimate of $\mathbf{w}_{M+}$ into the $\mathcal{P}-norm^{+}$ space. Then, we clip the values that are out of a certain range. In our experiments, we used the range $[-5\sigma, 5\sigma]$ as suggested in [33] where $\sigma$ is the standard deviation of in-domain latent codes. We then transform the clipped values back into the $\mathcal{W}^{+}$ space. We refer the readers to [33] for more details.

### 4.3. Regularization on Base Code $\mathbf{f}$

While optimizing Eq. (2) with the regularization on $\mathbf{w}_{M+}$ results in an in-domain solution for $\mathbf{w}_{M+}$, it still produces an improper solution for $\mathbf{f}$ that results in the failure of semantic image editing. Fig. 4(c) shows an inversion result using the reconstruction loss with the regularization on $\mathbf{w}_{M+}$. Thanks to the hard clipping in the $\mathcal{P}-norm^{+}$ space, the estimated $\mathbf{w}_{M+}$ is always in a desired range. However, the estimated $\mathbf{f}$ is still out-of-domain, and produces an incorrect style mixing result in Fig. 4(g).

To overcome this, we introduce a regularization method that encourages $\mathbf{f}$ to be in the extended domain of $\mathbf{f}$ defined in Sec. 3. Our method is a two-step approach. For an input image $I$, we first find an initial base code $\mathbf{f}^{o}$ that lies in the extended domain of $\mathbf{f}$ using an encoder $E$. Then, while optimizing Eq. (2), we find a base code $\mathbf{f}$ that is close to $\mathbf{f}^{o}$. To achieve this, we define a regularization loss for $\mathbf{f}$ as:

$$L_{\mathbf{f}}(\mathbf{w}^{*}) = \|\mathbf{f}^{o} - \mathbf{f}\|^{2} \tag{3}$$

where $\mathbf{f}^{o} = E(I)$.

Our final objective function is then defined as:

$$L(\mathbf{w}^{*}) = L_{recon}(\mathbf{w}^{*}) + \omega_{\mathbf{f}} L_{\mathbf{f}}(\mathbf{w}^{*}) \tag{4}$$

where $\omega_{\mathbf{f}}$ is a weight for the regularization loss $L_{\mathbf{f}}$. Our final approach optimizes Eq. (4) with the regularization on $\mathbf{w}_{M+}$. Fig. 4(d) and (h) show that our final approach can successfully invert an out-of-range image and support semantic image editing, respectively.

### 4.4. Encoder for Base Code $\mathbf{f}$

Our encoder estimates an initial base code $\mathbf{f}^{o}$ of an input image. As $\mathbf{f}^{o}$ has a small spatial resolution, e.g., $16 \times 16$, the encoder does not require an input image of the original resolution or a heavy network architecture. Thus, the encoder is designed to take a downsampled image of the resolution $8\times$ larger than $\mathbf{f}$, e.g., $128 \times 128$. The encoder has a VGG-like architecture [25] consisting of 11 convolution blocks and three pooling layers without fully connected layers. More details can be found in the supplementary material.

For the training of the encoder, we randomly sample a batch of latent codes from the latent space $\mathcal{Z}$ at each iteration. From each sampled latent code $\mathbf{z}$, we obtain its corresponding latent code $(\mathbf{f}^{gt}, \mathbf{w}^{gt}_{M+})$ and its image $I$. Using the sampled latent codes and their images, we train our encoder with a loss function defined as:

$$\begin{aligned} L_{enc} &= \left\|G(E(I_{\downarrow}), \mathbf{w}^{gt}_{M+}) - I\right\|^{2} \\ &+ \lambda_{per}\left\|F(G(E(I_{\downarrow}), \mathbf{w}^{gt}_{M+})) - F(I)\right\|^{2} \end{aligned} \tag{5}$$

where $I_{\downarrow}$ is a downsampled version of $I$. The first and second terms on the right-hand side are a MSE loss and a perceptual loss. The loss minimizes the difference between the training image $I$ and its reconstructed image using the latent code obtained by the encoder.

As we have $\mathbf{f}^{gt}$, we may use a loss term based on the distance between $E(I_{\downarrow})$ and $\mathbf{f}^{gt}$, e.g, $\|E(I_{\downarrow}) - \mathbf{f}^{gt}\|^{2}$. However, we found that using it instead of the loss terms in Eq. (5) leads to less accurate reconstruction of an input image.

Our training procedure does not use geometrically transformed images. Nevertheless, our encoder still performs effectively for geometrically transformed images thanks to the spatially-invariant property of CNNs. For example, for a shifted image, our encoder estimates a shifted feature map $\mathbf{f}^{o}$ that lies in the extended domain of $\mathbf{f}$.

Although Eq. (5) does not have any terms to encourage to predict a latent code in the extended domain, our encoder can effectively find a latent code that supports semantic image editing. As the encoder is trained using a large amount of images with a large batch size, we found that it is not necessary to include any other constraints such as the loss term based on the latent code distance.

## 5. Experiments

**Implementation details** In our implementation, we downsample images to $256 \times 256$ to compute the perceptual losses in $L$ and $L_{enc}$ following previous works [1, 19, 33]. In our experiments, we set $\omega_{per} = 10$, $\omega_{\mathbf{f}} = 10$ and

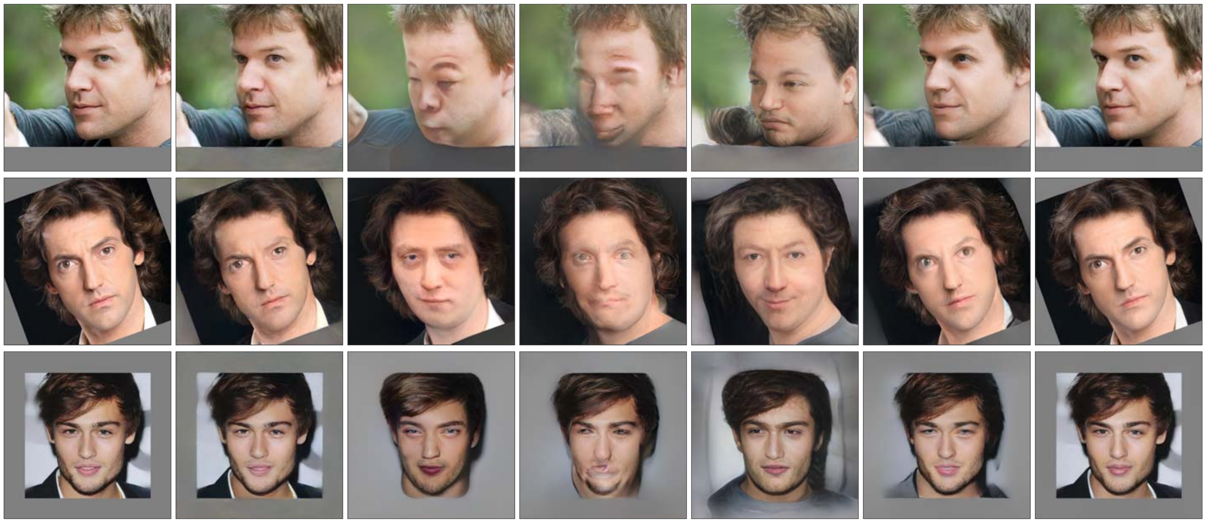

(a) Input (b) Im2StyleGAN (c) StyleGAN2 (d) P-*norm* (e) PSP (f) Ours (8×8) (g) Ours (16×16)

Figure 5: Qualitative comparison of the reconstruction quality of different methods. The input images are sampled from the CelebA-HQ dataset and applied different geometric transformations. Top to bottom: translation by 150 pix., rotation by 20 deg., and scaling by 3/4.

| | | | Translation | | | | Rotation | | | Scaling | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Models | Metric | | 0 | 50 | 100 | 150 | 10 | 20 | 30 | 7/8 ↓ | 3/4 ↓ | 9/8 ↑ | 5/4 ↑ |
| Im2StyleGAN [1] | PSNR | ↑ | 25.63 | 25.06 | 24.53 | 23.92 | 25.76 | 24.65 | 23.87 | 25.82 | 25.25 | 26.17 | 26.27 |
| | FID | ↓ | 48.37 | 45.73 | 52.52 | 58.64 | 50.06 | 56.63 | 65.76 | 33.80 | 34.24 | 38.02 | 36.78 |
| P-norm$^+$ [33] | PSNR | ↑ | 21.79 | 20.94 | 19.78 | 18.54 | 20.70 | 18.91 | 17.93 | 21.53 | 19.41 | 22.07 | 21.85 |
| | FID | ↓ | 58.69 | 64.52 | 78.56 | 98.53 | 77.93 | 86.16 | 110.48 | 46.89 | 60.38 | 52.76 | 49.06 |
| StyleGAN2 inv. [16] | PSNR | ↑ | 18.73 | 18.29 | 17.31 | 16.71 | 17.95 | 17.22 | 16.02 | 18.65 | 18.43 | 19.12 | 19.43 |
| | FID | ↓ | 65.49 | 70.36 | 78.32 | 87.70 | 79.31 | 82.25 | 96.23 | 52.26 | 50.23 | 60.64 | 60.24 |
| PSP [22] | PSNR | ↑ | 20.54 | 19.03 | 17.59 | 16.50 | 19.14 | 17.78 | 16.99 | 19.02 | 17.78 | 20.63 | 20.15 |
| | FID | ↓ | 78.53 | 84.85 | 99.66 | 118.50 | 108.13 | 115.46 | 142.09 | 84.87 | 96.29 | 70.16 | 68.32 |
| Ours ($8 \times 8$) | PSNR | ↑ | 23.69 | 23.35 | 23.74 | 23.50 | 23.30 | 22.06 | 21.35 | 23.37 | 22.72 | 23.93 | 24.22 |
| | FID | ↓ | 49.68 | 49.47 | 46.05 | 49.00 | 60.84 | 60.52 | 71.71 | 37.51 | 38.34 | 44.11 | 37.43 |
| Ours ($16 \times 16$) | PSNR | ↑ | **26.47** | **26.30** | **26.37** | **26.43** | **26.48** | **26.49** | **26.33** | **26.44** | **26.28** | **26.98** | **27.26** |
| | FID | ↓ | **30.27** | **32.16** | **30.68** | **31.58** | **37.01** | **33.96** | **33.98** | **24.92** | **24.29** | **27.61** | **23.84** |

Table 1: Quantitative comparison of the reconstruction quality of different methods on geometrically transformed images. For the evaluation, we sample 50 images from the CelebA-HQ dataset [14] and applied different degrees of translation, rotation and scaling.

| | | | Dataset | | |
|---|---|---|---|---|---|
| Models | Metric | | Bedroom | Tower | Cat |
| Im2StyleGAN | PSNR | ↑ | 19.88 | **20.64** | 22.90 |
| | FID | ↓ | 111.73 | 58.14 | 71.19 |
| IDinvert | PSNR | ↑ | 19.27 | 20.02 | - |
| | FID | ↓ | 80.21 | 75.59 | - |
| Ours (16x16) | PSNR | ↑ | **20.21** | 20.37 | **24.67** |
| | FID | ↓ | **49.92** | **42.89** | **31.74** |

Table 2: Quantitative comparison of the reconstruction quality of different methods on natural images. Each test set consists 25 images collected from the internet. For the bedroom and tower test sets, we use StyleGAN [15] models pre-trained on the LSUN bedroom and tower datasets [29]. For the cat test set, we use a StyleGAN2 [16] model pre-trained on the LSUN cat dataset. For the cat dataset, the results of IDinvert [31] are not available as IDinvert does not provide pre-trained weights for its encoder network.

$\lambda_{per} = 10$. For training the encoder, we set the batch size to 16 and the number of iterations to 10,000. We initially set the learning rate to 0.001 and reduced it by a factor of 0.1 every 2,000 iterations. For the inversion, we use 1,200 iterations with learning rate of 0.01. We use the Adam optimizer [18] both for the training of the encoder and GAN inversion. We conducted our experiments using pre-trained models of StyleGAN[1] and StyleGAN2[2].

In our experiments, we implement semantic editing operations by adding a semantic editing vector to a latent code, i.e., $\mathbf{w}^{edit} = \mathbf{w} + \alpha\mathbf{v}$ where $\alpha$ is a user parameter to control the editing strength and $\mathbf{v}$ is an editing vector following [23, 24, 31]. Specifically, we use editing vectors provided by IDinvert [31] and SeFa [24] for StyleGAN [15] and StyleGAN2 [16], respectively. For a latent code in $\mathcal{F}/\mathcal{W}^+$, we add an editing vector only to a detail code $\mathbf{w}_{M+}$.

**Reconstruction comparison** We first compare the reconstruction quality of our method with those of previous state-of-the-art inversion methods on the CelebA-HQ dataset [15]

[1] https://github.com/genforce/idinvert_pytorch

[2] https://github.com/genforce/genforce

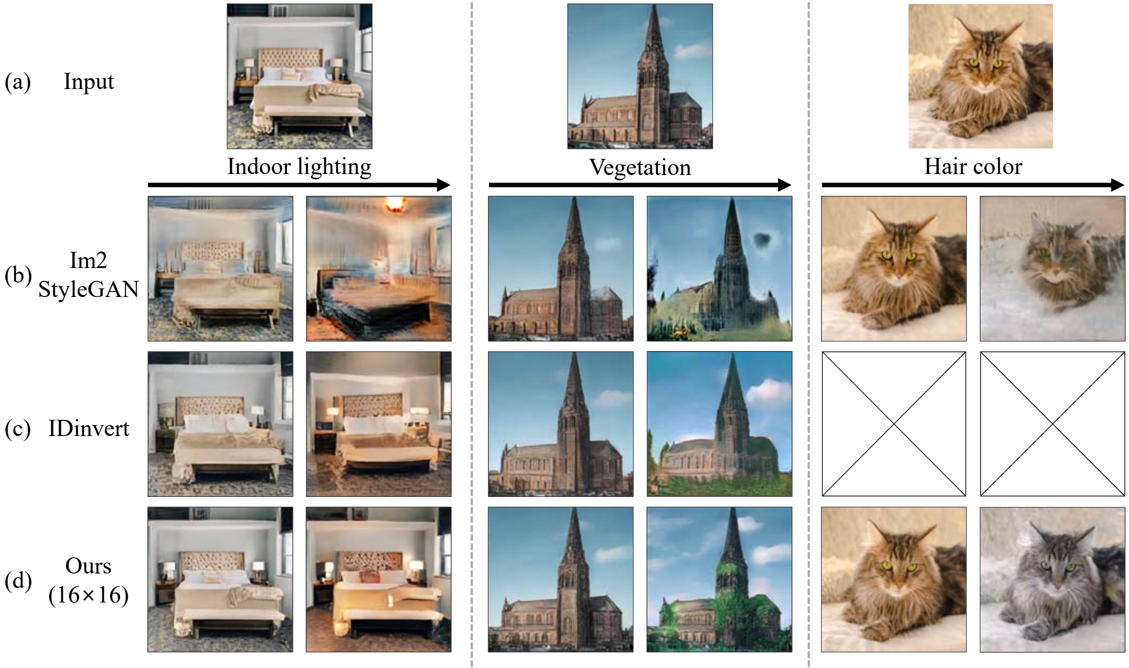


Figure 6: Qualitative comparison of the reconstruction and semantic editing quality of different methods on natural images. The input images on the top row are collected from the internet. For the bedroom and tower image on the left and middle, we use StyleGAN [15] models pre-trained on the LSUN bedroom and tower datasets [29]. For the cat image on the right, we use a StyleGAN2 [16] model pre-trained on the LSUN cat dataset. For the cat dataset, the results of IDinvert [31] are not available as IDinvert does not provide pre-trained weights for its encoder network. From left to right, the semantic editing operations are indoor lighting, vegetation and hair color change.

using a StyleGAN2 [16] model pre-trained on the FFHQ dataset [15]. For the comparison, we constructed a test set composed of 50 images randomly extracted from the CelebA-HQ dataset [14]. In order to investigate the inversion performance on out-of-range images with geometric transformations, we applied different transformations to the test set. Specifically, we applied translation of 50, 100, and 150 pixels in random directions, rotation by 10, 20, and 30 degrees randomly in a counterclockwise and clockwise direction, and scaling by 7/8, 3/4, 9/8, and 5/4.

We compare our method with state-of-the-art methods: Im2StyleGAN [1], StyleGAN2 inversion [16], P-norm$^+$ [33], and PSP [22]. PSP is an encoder-based method while the others are optimization-based ones. We used the authors' code for StyleGAN2 and PSP. We implemented Im2StyleGAN and P-norm$^+$ as their code is not available. We also compare two versions of our method, which use a base code **f** of size $8 \times 8$ and $16 \times 16$, respectively.

Fig. 5 shows a qualitative comparison. As shown in the figure, all the methods except for Im2StyleGAN [1] and ours fail to reconstruct the input images. Table 1 reports a quantitative comparison in PSNR and FID [11]. We refer the readers to our supplementary material for additional comparison in SSIM [27] and RMSE. The table shows that our $16\times16$ version achieves the highest reconstruction quality both in PSNR and FID for all geometric transformations. Both in the figure and table, Im2StyleGAN shows high-quality reconstruction results. However, due to the lack of in-domain constraints, Im2StyleGAN tends to produce out-of-domain latent codes that are not semantically editable as will be seen later in this section. The table also shows that the performances of the previous methods degrade quickly for larger translations and rotations. For example, the performance of P-norm$^+$ [33] drops by 3.86 dB for the rotation by 30 degrees. Our $8 \times 8$ version performs worse than the $16 \times 16$ version as it uses a more constrained latent space. We also note that our $16 \times 16$ version outperforms all the other methods even for images without geometric transformations (Translation = 0 in Table 1) thanks to the base code **f** supporting local variations.

**Inversion of natural images** Due to the large diversity of natural images, it is difficult to accurately reconstruct and edit a natural image using previous GAN inversion approaches. On the other hand, thanks to the high degree-of-freedom of the $\mathcal{F}/\mathcal{W}^+$ space, our approach is especially effective in handling natural images. To verify this, we compare the reconstruction and editing quality of previous methods and ours on natural images. For evaluation, we use StyleGAN and StyleGAN2 models [15, 16] pre-trained on the LSUN bedroom, tower and cat datasets [29]. We also collected 25 bedroom, tower and cat images each from the internet and used them as our test sets so that the images in

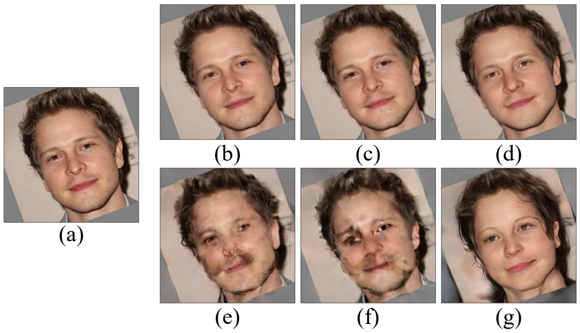


Figure 7: Ablation study. (a) Target image. (b) Reconstruction using only the reconstruction loss. (c) Reconstruction using the regularization on $\mathbf{w}_{M+}$. (d) Reconstruction using the regularization both on $\mathbf{w}_{M+}$ and $\mathbf{f}$ (our final method). Semantic editing results of (b), (c) and (d) are shown in (e), (f) and (g), respectively.

the test sets are of the same classes as the training images of the pre-trained models, but not aligned with the training images. For these datasets, we use 3,000 iterations.

We compare our method against Im2StyleGAN [1], which shows high-quality reconstruction results in the previous experiment, and IDinvert [31], which finds an in-domain latent code for semantic editing. We use the authors' code for IDinvert. Fig. 6 shows a qualitative comparison of the reconstruction and editing qualities. Both Im2StyleGAN [1] and IDinvert [31] produce less accurate reconstruction results than ours. Their editing results also show artifacts due to the out-of-range input images. Especially, the editing results of Im2StyleGAN have severe artifacts as its out-of-domain latent codes. In contrast, our method shows high-quality reconstruction and editing results for all three cases. Table 2 shows a quantitative comparison of the reconstruction qualities. The table also shows that our method achieves high reconstruction quality on natural images compared to the other methods. More results can be found in the supplementary material.

**Ablation study** Fig. 7 shows a qualitative comparison of variants of our method using StyleGAN2 [16] to verify the effectiveness of our regularization scheme. While all the variants show excellent reconstruction results thanks to the high degree of freedom of the $\mathcal{F}/\mathcal{W}^+$ space, the editing results of the variants that use only the reconstruction loss or regularization on the detail code $\mathbf{w}_{M+}$ are severely degraded. On the other hand, the editing result of our final model in (d) looks the most natural thanks to our regularized inversion scheme. More examples and a quantitative evaluation are in the supplementary material.

**Editing operations *v.s.* scale of base code f** Finally, we analyze the effect of the scale of the base code $\mathbf{f}$ on image editing. Using a feature map at a finer-scale for the base code $\mathbf{f}$ leads to higher reconstruction quality as shown in Fig. 5 and Table 1. On the other hand, it also reduces the diversity of semantic editing operations. Especially, it makes it difficult to perform semantic operations that rely on coarse-scale latent codes $\mathbf{w}_i$ in the $\mathcal{W}$ space. Fig. 8 shows an example. While our method with $\mathbf{f}$ of size $8 \times 8$ supports both pose changing and aging, ours with $\mathbf{f}$ of size $16 \times 16$ does not support pose changing since the pose changing operation requires to edit small-scale latent codes.

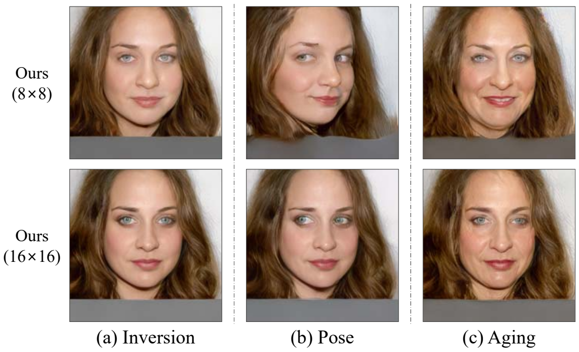


Figure 8: Available editing operations according to the size of $\mathbf{f}$. (a) shows inversion results using $\mathbf{f}$ of different sizes. (b) and (c) show results of different editing operations. The pose editing operation that changes the overall image structure does not work for $\mathbf{f}$ of size $16 \times 16$ while the aging operation works for both.

## 6. Conclusion

In this paper, we proposed *BDInvert*, a novel GAN inversion approach for semantic editing of out-of-range images with geometric transformations. Based on the StyleGAN and StyleGAN2 frameworks [15, 16], we presented an alternative latent space $\mathcal{F}/\mathcal{W}^+$ that supports geometric transformations of an image as well as its semantic manipulation. To find a proper solution in the $\mathcal{F}/\mathcal{W}^+$ space that is semantically editable, we introduced a novel regularized optimization approach. We verified the effectiveness of our approach both qualitatively and quantitatively.

**Limitations** As discussed in Secs. 3 and 5, the $\mathcal{F}/\mathcal{W}^+$ space reduces the diversity of semantic editing operations. Also as our approach is based on optimization, it requires a relatively long computation time. With an Nvidia RTX 3090 GPU, it takes about 3 minutes for a $1024 \times 1024$-sized image. Our approach cannot handle images with severe geometric transformations. However, this can be easily resolved by rough alignment of an input image as our method does not require accurate alignment. Finally, our method cannot handle images that are too different from the training dataset. See the supplementary material for examples.

**Acknowledgements** This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)) and National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (NRF-2018R1A5A1060031, No. 2020R1C1C1014863).

# References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *CVPR*, 2019. 1, 2, 3, 5, 6, 7, 8

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020. 2

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 38(4), July 2019. 1, 2

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *ICLR*, 2019. 1

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 1, 2

[6] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 2, 4

[7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NIPS*, 2014. 1

[8] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020. 2, 4

[9] 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. 2

[10] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020. 1, 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 7

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[13] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *ECCV*, 2020. 2

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1, 6, 7

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[17] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, pages 1867–1874, 2014. 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4, 6

[19] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, 2020. 5

[20] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, 2020. 2

[21] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1

[22] 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 *CVPR*, 2021. 1, 2, 6, 7

[23] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 1, 2, 6

[24] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021. 2, 6

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[26] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 1, 2

[27] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[28] 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. In *CVPR*, 2017. 2

[29] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6, 7

[30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[31] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020. 1, 2, 3, 4, 6, 7, 8

[32] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 2

[33] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*, 2020. 1, 2, 5, 6, 7