Surrogate Gradient Field for Latent Space Manipulation

Minjun Li* Yanghua Jin* Huachun Zhu
Preferred Networks
{minjunli, jinyh, zhu}@preferred.jp

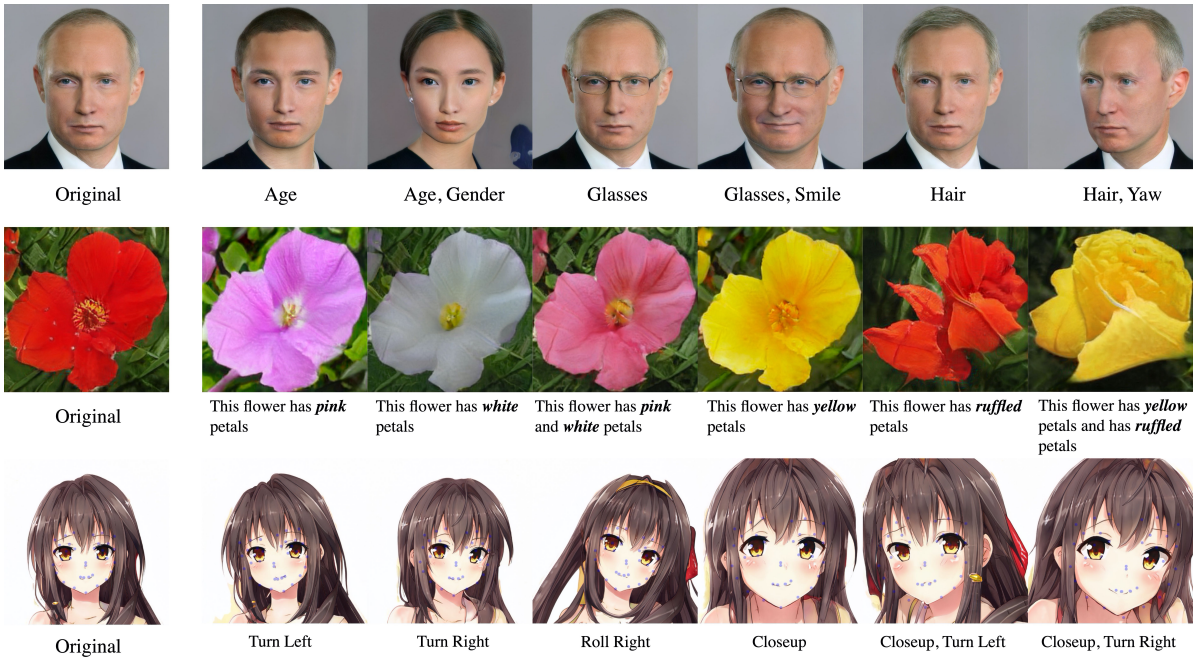

Figure 1. **Our Surrogate Gradient Field (SGF) can edit images with diverse modalities of control by manipulating the latent codes of GANs.** In the first row, we adjust the facial attributes of a person's photo. In the second row, we use natural language sentences to alter the color and the shape of a generated flower. In the last row, we edit keypoints of a generated anime character to modify head poses. We use StyleGAN2 [20] to generate the images above.

Abstract

Generative adversarial networks (GANs) can generate high-quality images from sampled latent codes. Recent works attempt to edit an image by manipulating its underlying latent code, but rarely go beyond the basic task of attribute adjustment. We propose the first method that enables manipulation with multidimensional condition such as keypoints and captions. Specifically, we design an algorithm that searches for a new latent code that satisfies the target condition based on the Surrogate Gradient Field (SGF) induced by an auxiliary mapping network. For quantitative comparison, we propose a metric to evaluate the disentanglement of manipulation methods. Thorough experimental analysis on the facial attribute adjustment task shows that our method outperforms state-of-the-art methods in disentanglement. We further apply our method to tasks of various condition modalities to demonstrate that our method can alter complex image properties such as keypoints and captions.

1. Introduction

Generative Adversarial Networks [9], or GANs, are one of the most popular and effective methods for generating high fidelity images. In the simplest form, the generator model creates a random image from a latent code sampled from the latent space. To create an image that matches some target properties, however, we need a method to condition

*Equal contribution.

the generated image on such properties. In other words, the method should be able to incorporate a piece of information, such as attributes, keypoints, or even an interpretation of the image in a natural language, into the generation of the image. Intuitively, to condition the image, we can instead condition its latent code on the same information, in an attempt to generate an image that satisfies the target properties.

As an increasingly popular approach to image modification [1] and GAN interpretation [31], **latent space manipulation** is a type of approach that bases on varying the latent codes of images. The generator maps manipulated latent codes to images that hopefully match target properties. To be specific, InterFaceGAN [31] and GANSpace [11] find meaningful directions in latent space, and vary latent codes along these directions to adjust the attributes of images.

Although existing methods explore the potential application of latent space manipulation, these methods still suffer from the following limitations. To begin with, the disentanglement of manipulation can be limited. Adjustment of one attribute of an image is occasionally accompanied by some undesirable shifts in other attributes. Moreover, existing methods are restricted to one-dimensional conditioning. In other words, these methods excel in adjusting attributes such as smiling or not, female or male, each of which can be parameterized by a scalar condition. However, these methods do not provide a general solution to complex modifications that condition on multidimensional information (*e.g.* the pose of a human or the caption of an image).

We suggest that there is another line of latent space manipulation based on optimization. Using a generator and an image classifier, we can optimize the latent code for minimizing the difference between the properties of the current image and the target properties. Empirically, this simple approach does not work as expected because both the classifier and the generator are highly non-convex deep neural networks. As a result, the gradient field in the latent space may be misleading, and thus the optimization of a latent vector is often trapped in a local optimum.

To overcome the difficulty of the optimization-based approach, we propose a novel method for latent space manipulation. In our method, we train an auxiliary mapping network that induces a Surrogate Gradient Field (SGF). We design an algorithm that uses SGF in search of a new latent code that satisfies a target condition. For comparison with existing works, we design a metric that evaluates the disentanglement of a manipulation method. Based on the metric, we conduct thorough quantitative experiments and a user study to demonstrate that our method outperforms state-of-the-art methods in the disentanglement of manipulation. As the first work towards multidimensional conditioning with latent space manipulation, our method successfully modifies images utilizing keypoints and captions, illustrated with qualitative results.

To summarize our main contributions,

- We propose the first latent space manipulation method of GANs that supports multidimensional conditioning.
- We conduct quantitative experiments and a user study on the task of facial attribute adjustment to demonstrate that our method outperforms state-of-the-art methods in disentanglement.
- We apply our method to latent space manipulation using keypoints and captions, justifying our method as a unified approach for various modalities of conditioning.

2. Related work

Generative Adversarial Networks. GAN [9] has shown great potential on generating photo-realistic images [27, 18]. It has been applied to a wide range of tasks including image editing [4, 31], image translation [15, 36] and super-resolution [22]. Recent works have made tremendous progress on generating high-quality photo-realistic image [3, 10, 6, 18, 19]. Among the existing works on image generation, one of the most well-known works is StyleGAN [19] which introduces a stacked architecture that enables high-resolution image generation with fine-grained control. Its recent follow-up work StyleGAN2 [20] further improved the generated image qualities and achieved state-of-the-art image synthesis results. Our work greatly benefits from the progress of the GAN because we can apply our method to various GAN models.

Manipulation on Latent Vector. Early GAN works [27] have already discovered that generated images can be semantically edited by applying vector arithmetic on the latent space. Since vector arithmetic-based approach is straightforward and model agnostic, recent works continue to explore in this direction. Existing methods can be categorized into two classes: supervised methods [31, 26, 8] and unsupervised methods [11, 32]. Supervised methods use an extra classifier to label properties of generated images. Shen *et al.* [31] train a linear SVM on pairs of latent vectors and labels to find a decision hyperplane. Latent vectors are then moved along the normal direction of the decision hyperplane for adjusting attributes. For multiple attributes, their method can sacrifice performance for disentanglement by orthogonalizing each direction vector. On the other hand, unsupervised methods directly find semantically meaningful directions by PCA [11] or self-supervised learning [32]. Besides vector arithmetic-based approaches, some more recent works [16, 2] introduce non-linear transformations and generative modelings in the latent space to adjust multiple attributes simultaneously.

In contrast with existing methods, our approach utilizes a neural network to model complicated semantic relationships between latent vectors and corresponding predictions.

We further extend the scope of conditions to a wider variety of vector representations. We show that our method achieve a higher degree of disentanglement compared with other methods.

3. Method

3.1. Problem Definition

Let $G\colon \mathcal{Z} \to \mathcal{X}$ be a pretrained GAN generator. $\mathcal{Z} \subseteq \mathbb{R}^d$ is the d-dimensional latent space [*], and $\mathcal{X}$ denotes the space of generated image. The classifier network $C\colon \mathcal{X} \to \mathcal{C}$ predicts semantic properties $c \in \mathcal{C} \subseteq \mathbb{R}^{n_c}$ from a generated image $x \in \mathcal{X}$. Although C can be as simple as a multi-label classifier, where $\mathbb{R}^{n_c}$ stands for the space of n_c semantic attributes, the setting actually applies to any embedding in Euclidean space. For example, keypoints detector with n_p points on a 2D image can be regarded as an embedding to $\mathbb{R}^{2n_p}$.

Define $\Phi(z) \coloneqq C(G(z))$ for convenience. Suppose we have a latent vector $z_0 \in \mathcal{Z}$, its corresponding properties $c_0 = \Phi(z_0)$ and target properties c_1. Our goal is to find $z_1 \in \mathcal{Z}$ such that $\Phi(z_1) = c_1$.

3.2. Learning the Auxiliary Mapping

A powerful generator such as StyleGAN2 [20] may easily generate infinite images that match the properties c_1. We would like to attain the desired properties c_1 with minimal unwanted modification to the image. Intuitively, in $\mathcal{Z}$ space, z_0 can be slightly perturbed to get a z_1 that is sufficiently close to z_0. Empirically, the gradient field of Φ is not suitable for perturbing z_0, so we seek to replace it with a new gradient field.

As a preparation, we introduce an auxiliary mapping $F : \mathcal{Z} \times \mathcal{C} \to \mathcal{Z}$ satisfying

$$F(z, \Phi(z)) = z, \forall z \in \mathcal{Z} \tag{1}$$

In our implementation, F is a multi-layer neural network, and trained using a simple reconstruction loss. Inspired by Behrmann *et al.* [5], we use spectral normalization [24] in F so that its Lipschitz constant $\mathrm{Lip}(F) < 1$. As a result, the operator norm of its Jacobian is less than 1 [12]. Furthermore, for any eigenvalue λ_F of the Jacobian of F and the corresponding unit eigenvector x_F, we have $\|\lambda_F x_F\| = \left\|\frac{\partial F(z,c)}{\partial z} x_F\right\| \le \left\|\frac{\partial F(z,c)}{\partial z}\right\|_{\mathrm{op}} < 1$, where $\|\cdot\|_{\mathrm{op}}$ denotes operator norm. Therefore, the spectral radius of the Jacobian of F satisfies

$$\rho\left(\frac{\partial F(z,c)}{\partial z}\right) \le \left\|\frac{\partial F(z,c)}{\partial z}\right\|_{\mathrm{op}} < 1 \tag{2}$$

[*]For StyleGAN, a latent vector z is first sampled from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ in Z-space, and a fully-connected neural network then transforms it into a new latent vector w in W-space. In our formulation, $\mathcal{Z}$ can be either Z-space or W-space.

Figure 3 shows the training pipeline of F.

3.3. Manipulation with Surrogate Gradient Field

To formalize the perturbation of z_0, we define a path $z(t), t \in [0, 1]$ in the latent space that starts from z_0 and ends at z_1, i.e. $z(0) = z_0$ and $z(1) = z_1$. Here we make several assumptions about path $z(t)$.

1. The generator is capable of generating an image that match the desired properties:

$$\exists\, z_1 \in \mathcal{Z} \quad s.t. \quad \Phi(z_1) = c_1$$

2. While traversing the path, the properties $\Phi(z(t))$ of the generated image changes at a constant rate, i.e.

$$\frac{\mathrm{d}\Phi(z(t))}{\mathrm{d}t} = c_1 - c_0 \tag{3}$$

3. $\forall\, z \in \mathcal{Z}$,

$$\frac{\partial F(z, \Phi(z))}{\partial c} \neq 0 \tag{4}$$

The assumptions above suggests that 1. our task is well-posed, 2. path $z(t)$ is a smooth interpolation between the original properties and the target properties, and 3. F is not a trivial mapping that just map any (z, c) pair to z.

Now we derive the surrogate gradient field of Φ. Using Eq. (1) of auxiliary mapping F, we can rewrite the path as

$$z(t) = F(z(t), \Phi(z(t))) \tag{5}$$

Take time derivatives on both sides, we have

$$\begin{aligned}
&\frac{\mathrm{d}z(t)}{\mathrm{d}t} = \frac{\mathrm{d}F(z(t), \Phi(z(t)))}{\mathrm{d}t} \\
&= \frac{\partial F(z(t), \Phi(z(t)))}{\partial z}\frac{\mathrm{d}z(t)}{\mathrm{d}t} + \frac{\partial F(z(t), \Phi(z(t)))}{\partial c}\frac{\mathrm{d}\Phi(z(t))}{\mathrm{d}t} \\
&= \frac{\partial F(z(t), \Phi(z(t)))}{\partial z}\frac{\mathrm{d}z(t)}{\mathrm{d}t} + \frac{\partial F(z(t), \Phi(z(t)))}{\partial c}(c_1 - c_0)
\end{aligned}$$

We plug in assumption 2 in the last step. Organize $\frac{\mathrm{d}z(t)}{\mathrm{d}t}$ to the left hand side and rearrange the last equation, we have $\frac{\mathrm{d}z(t)}{\mathrm{d}t} = \left(\mathbf{I} - \frac{\partial F(z(t), \Phi(z(t)))}{\partial z}\right)^{-1} \frac{\partial F(z(t), \Phi(z(t)))}{\partial c}(c_1 - c_0)$, the invertibility implied by Eq. (2) [12].

Define surrogate gradient field H as

$$H(z) \coloneqq \left(\mathbf{I} - \frac{\partial F(z, \Phi(z))}{\partial z}\right)^{-1} \frac{\partial F(z, \Phi(z))}{\partial c}(c_1 - c_0) \tag{6}$$

Note that $H(z) \neq 0, \forall z \in \mathcal{Z}$ because of Eq. (2) and assumption 3. We arrive at our ordinary differential equation,

$$\begin{cases} \frac{\mathrm{d}z(t)}{\mathrm{d}t} &= H(z(t)),\ t \in [0, 1] \\ z(0) &= z_0 \end{cases} \tag{7}$$

Algorithm 1 Manipulating GAN with surrogate gradient field

Input: Generator G, Classifier C, auxiliary mapping F, order of the series expansion m, iteration number n, initial latent vector z_0, target attributes c_1, step size λ

$c_0 \leftarrow C(G(z_0))$
$\delta_c \leftarrow \lambda(c_1 - c_0)$
$c^{(0)} \leftarrow c_0$
for $i = 1, \cdots, n$ **do**
  $\delta_z^{(0)} \leftarrow \frac{\partial F}{\partial c}(z^{(i-1)}, c^{(i-1)})\delta_c$
  $\delta_z \leftarrow \delta_z^{(0)}$
  for $j = 1, \cdots, m$ **do**
    $\delta_z^{(j)} \leftarrow \frac{\partial F}{\partial z}(z^{(i-1)}, c^{(i-1)})\delta_z^{(j-1)}$
    $\delta_z \leftarrow \delta_z + \delta_z^{(j)}$
  end for
  $z^{(i)} \leftarrow z^{(i-1)} + \delta_z$
  $c^{(i)} \leftarrow C(G(z^{(i)}))$
  if $c^{(i)}$ close to c_1 **then**
    return $z^{(i)}$
  end if
end for
return $z^{(n)}$

Figure 2. **Pseudocode of our manipulation algorithm.** The outer loop is a simple forward Euler ODE solver, which computes the movement δ_z, and accumulate to the current latent vector $z^{(i)}$. The classifier predicts the properties of image at each time step to determine when to stop. The inner loop approximates the matrix inversion term in Eq. (6) using the Neumann series.

3.4. Numerical Solution of the ODE

To compute our goal $z(1)$, we solve the initial value problem (Eq. (7)) using a numerical ordinary differential equation solver. Nevertheless, it is time consuming and potentially numerically unstable to calculate the Jacobian of F and the matrix inversion when evaluating $H(z)$ (Eq. (6)). Instead, we apply Neumann series expansion [12] to approximate the matrix inversion. For a matrix $\mathbf{X}$ that satisfies $\rho(\mathbf{X}) < 1$, the following expansion converges

$$(\mathbf{I} - \mathbf{X})^{-1} = \mathbf{I} + \mathbf{X} + \mathbf{X}^2 + \dots$$

Another obstacle to numerical computation is that, in reality, the path may deviates from the assumption 2. To be specific, at step i with a step size of λ, $\Phi(z(i\lambda))$ does not precisely equals $\Phi(z((i-1)\lambda)) + \lambda(c_1 - c_0)$. Two source of error leads to the problem: one from the numerical solver, and another from not having a perfect F which has $F(z, \Phi(z)) = z$ exactly everywhere. To overcome this difficulty, in practice we fix the step size λ but do not necessarily stop the iteration process at step $1/\lambda$. The algorithm checks the properties $c_i = \Phi(z(i\lambda))$ at each step, and stops

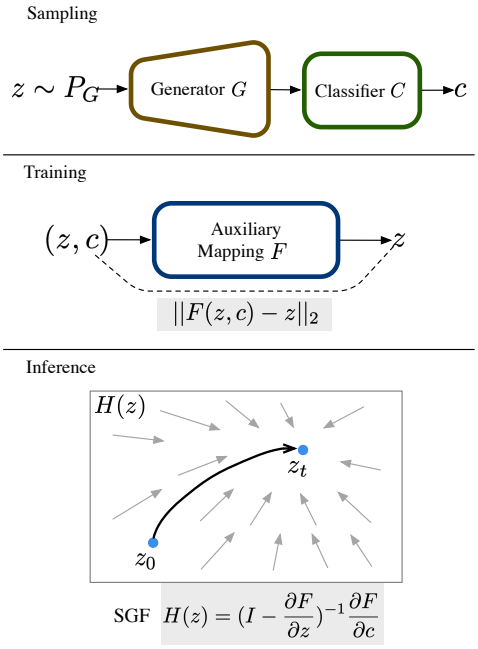

Figure 3. **Overview of our method.** P_G denotes the distribution of latent vectors in a latent space, which can be either Z-space or W-space in the case of StyleGAN. We sample (z, c) pairs, and train the auxiliary mapping F using MSE loss. The surrogate gradient field H navigates the latent vector to the target in the inference stage.

only when c_i is sufficiently close to the target c_1, unless it reaches the maximum step number. Algorithm 1 shows the summary of the manipulation procedure.

4. Experiments

4.1. Compared Methods

We compare the proposed method SGF with two state-of-the-art latent space manipulation methods: **InterfaceGAN** [31] [†] and **GANSpace** [11] [‡]. All compared methods are tested using the official code release.

InterfaceGAN. We retrain the InterfaceGAN model for each control attribute. Since InterfaceGAN can only learn one binary attribute at once, we train on each attribute independently with the same training data of our SGF strictly following the training setting in the paper.

GANSpace. For GANSpace, we use the pre-selected control vectors released in its official code and only apply changes to the recommended StyleGAN2 layers.

4.2. Generator Models and Datasets

Choosing different combinations of the latent space $\mathcal{Z}$ and condition space $\mathcal{C}$, we set up four distinct settings for

[†] https://github.com/genforce/interfacegan
[‡] https://github.com/harskish/ganspace

latent space manipulation to demonstrate that our method can control different generator models under various types of conditions.

For the generator, we test StyleGAN2 [20] and ProgressiveGAN [18]. StyleGAN2 experiments are conducted on W-space, while ProgressiveGAN experiments are conducted on Z-space. To further demonstrate that our method can accept various types of conditions besides image attributes, we conduct experiments on two other representative properties (*i.e.*, keypoints and image captions). We only show the results of our SGF method for keypoints and image captions, since other methods are not able to utilize these conditions.

FFHQ-Attributes. We adopt a pretrained FFHQ StyleGAN2 [20] as the generator for experiments on facial attributes editing. For the classifier, we fine-tune a pretrained SEResNet50 [13] model from VGGFaces2 [7] dataset. We construct the training data for the classifier model by labeling $100K$ randomly sampled images with the Azure Face API [§], and combine them with labeled faces from the CelebA [23] dataset. With duplicate labels removed, the final classifier can predict 48 facial attributes. Among them, we select four representative attributes, which includes both highly entangled attributes ("gender" and "bald") and less entangled ones ("smile" and "black hair"), for quantitative comparisons and the user study.

CelebAHQ-Attributes. To compare the performance on models other than StyleGAN, we also test a ProgressiveGAN [18] pretrained on the CelebAHQ dataset. We use the same facial attributes classifier as the FFHQ-Attributes in this experiment.

Anime-KeypointsAttr. We follow [17, 30] to build a high-quality Japanese anime-face dataset and train a StycleGAN2 on it. We base on the animeface-2009 [¶] and illustration2vec [30] to create facial landmarks keypoints and image attributes as the conditions for manipulation.

Flowers-Caption. Previous works have shown great success on training GANs conditioned on text captions [34]. However, to our best knowledge, SGF is the first method that can utilize text captions to conditionally manipulate latent vectors of a pretrained GAN. Our experiment is based on a pretrained image generator model [35] on Oxford-102 Flowers dataset [25]. The image caption generator is an attention-based caption model [33] trained on flower caption dataset [28]. To fit our pipeline for latent space manipulation, we use the sentence transformer [29] to encode generated captions into vectors.

[§]https://azure.microsoft.com/en-us/services/cognitive-services/face/
[¶]https://github.com/nagadomi/animeface-2009

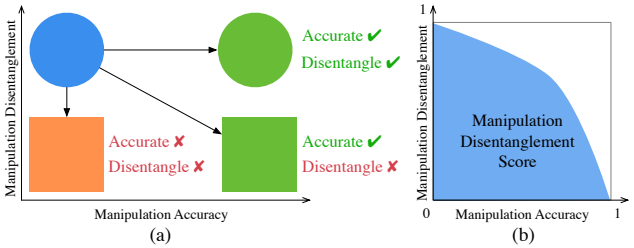

Figure 4. **Illustration of the Manipulation Disentanglement Score (MDS).** (a) Manipulating the color of a blue circle to green while keeping the shape unchanged. (b) MDS is defined as the AUC of Manipulation Disentanglement Curve (MDC).

4.3. Implementation Details

The auxiliary mapping F is implemented with an N-layer MLP combined with AdaIN [14]. We also apply spectral normalization [24] to all fully-connected layers. We describe the detail of network architectures in the supplementary material. For Z-space experiments, we set $N = 6$. While, for W-space experiments, we observe that F can easily degenerate to a trivial mapping by ignoring conditions c when $N = 6$. To prevent the degeneration, we increase N to 15 for all W-space experiments. For each experiment, we sample $200k$ pairs of latent vectors and corresponding conditions to build the training dataset of F. We apply a truncation rate of 0.8 to all StyleGAN2 samples. We train F for $500k$ iterations with a batch size of 8 using Adam optimizer [21] with learning rate of 0.0002. For the manipulation, we apply Algorithm 1 with order $m = 1$ and step size $\lambda = 0.2$ as default.

4.4. Evaluation Metrics

It is difficult to designing comprehensive quantitative metrics for measuring the disentanglement of latent space manipulation methods, which often use model-specific hyper-parameters to control the editing strength. For example, Figure 5(b) shows manipulation results of "gender" from different methods under different editing strength. Shen *et al.* [31] use the number of prediction changes to measure disentanglement among different attributes. However, comparing only the final results of image manipulation algorithms can be unfair. When editing strength increases, some methods tend to over-modify the image, i.e. introducing unwanted modification. Therefore, for comprehensive measurement of disentanglement, it is necessary to design an editing strength-agnostic metric.

4.4.1 Manipulation Disentanglement Score

For a given manipulation goal, a trade-off between accuracy and disentanglement often exists. Figure 4(a) illustrates the possible ways to change a blue circle to a green one. For a both accurate and disentangled manipulation, the color be-

comes green while the shape keeps round. An example of accurate but entangled manipulation would be changing the shape to a square when the color turns green.

By gradually increases manipulation strength and calculate the accuracy and disentanglement measure at each point, we can plot these points on the accuracy-disentanglement plane to attain a **Manipulation Disentanglement Curve** (MDC). As Figure 4(a) suggests, a method with an MDC closer to $y = 1$ indicates overall better disentanglement. In this way, we can compare the MDCs with each other in different methods.

In reminiscence of ROC curve, we define **Manipulation Disentanglement Score** (MDS) as the Area under Curve (AUC) of an MDC, illustrated in Figure 4(b). A method with a higher MDS suggests that it has a higher degree of disentanglement for the given manipulation.

For an experiment of attributes manipulation with N samples, suppose we can infer the scores of M attributes in total from an image. We consider an attribute is changed if the score changes more than 0.5 during the manipulation. Suppose there are N_s sample which successfully have their attributes changed to the target attributes. The manipulation accuracy is then the success rate N_s/N. For sample i, if n_i attributes *other than* the target attribute have changed, we can use $\frac{1}{N}\sum_{i=1}^{N}(1 - \frac{n_i}{M-1})$ as the manipulation disentanglement. An alternative way to define manipulation disentanglement is using image similarity, however, we found it less sensitive to subtle changes like added beards compared to the image attribute classifier we use. In our experiments on facial attributes manipulation, we evaluate $N = 100$ samples for each attribute, and $M = 48$. We inverse the direction of manipulation for samples that already match the target attribute so that we can calculate manipulation accuracy for every sample.

4.4.2 User Study

In addition to quantitative comparison on MDS, we conduct a user study in the FFHQ facial attributes experiments to further evaluate the disentanglement of methods. For each question of the user study, a user would see a source image and manipulation results from both our SGF and the InterfaceGAN. The user is then asked to choose a result that has best changed the source image to match a target attribute while keeping other features unchanged. We use 10 random generated images and 10 photos projected to the latent space of GAN [20]. In total, 20 participants have made 400 preference choices.

4.5. Comparisons on FFHQ-Attributes

Experiments on attributes manipulation compare SGF to the baseline models in the perspectives of manipulation disentanglement and accuracy defined in Sec. 4.4.1. In Figure 5(a), we plot the Manipulation Disentanglement Curves (MDCs) for our proposed SGF with state-of-the-art methods on four facial attribute editing settings. Our method has shown a better or comparable disentanglement degree compared with other methods.

Table 1. **MDS comparison on facial attribute editing on FFHQ-Attributes and CelebaHQ-Attributes.** Our SGF method shows the best overall score in attribute editing experiments on both FFHQ and CelebaHQ datasets, and significantly outperforms the compared methods on attributes that tend to be entangled (*e.g.* "gender" and "bald").

MDS on FFHQ-Attributes					
Method	**Gender**	**Bald**	**Smile**	**Black Hair**	**Overall**
GANSpace	0.841	0.491	0.248	0.543	0.531
InterfaceGAN	0.808	0.254	**0.883**	0.938	0.721
SGF (Ours)	**0.919**	**0.590**	**0.884**	**0.955**	**0.837**
MDS on CelebAHQ-Attributes					
Method	**Gender**	**Bald**	**Smile**	**Black Hair**	**Overall**
InterfaceGAN	0.876	0.442	0.856	0.876	0.758
SGF (Ours)	**0.912**	**0.799**	**0.896**	**0.897**	**0.876**

From the MDC of "gender" in baseline methods, we observe a sacrifice of manipulation disentanglement for high accuracy, which suggests that high manipulation strength in baseline methods introduces changes in non-target attributes . Figure 5(b) qualitatively compare the results of editing "gender" attribute. Our method changes "gender" without side effects such as adding beards. In contrast, both the InterfaceGAN and GANSpace add non-target properties to the final results when manipulation strength is high. We make the same observation on the "gender" MDC in Figure 5(a): as accuracy increases with the manipulation strength, the disentanglement degree of all methods except SGF drops significantly. This suggests that while accuracy of baseline methods comes at the price of entanglement, our method is able to achieve high accuracy and disentanglement at the same time.

In Figure 5(c), we qualitatively compare SGF with InterfaceGAN and GANSpace on editing other attributes. For each method and attribute, we use the hyper-parameters in settings highlighted with green circles in Figure 5(a). For each highlighted setting, the harmonic mean of accuracy and disentanglement reach the peak on the curve. while editing the target attribute, SGF consistently changes the least number of other properties. InterfaceGAN achieves similar disentanglement in "smile", while showing inferior results in both "bald" and "black hair". GANSpace shows inferior results in all settings.

We calculate the AUC for each method and attribute in Figure 5(a) as the MDS in Table 1. We find some attributes tend to correlate with others, *e.g.* "bald" often correlates with "gender" (Figure 5(b)). For experiments of such attributes, our proposed method significantly outperforms others. For editing relatively less entangled attributes, *e.g.* "smile" and "black hair", our method has comparable results with InterfaceGAN and outperforms GANSpace.

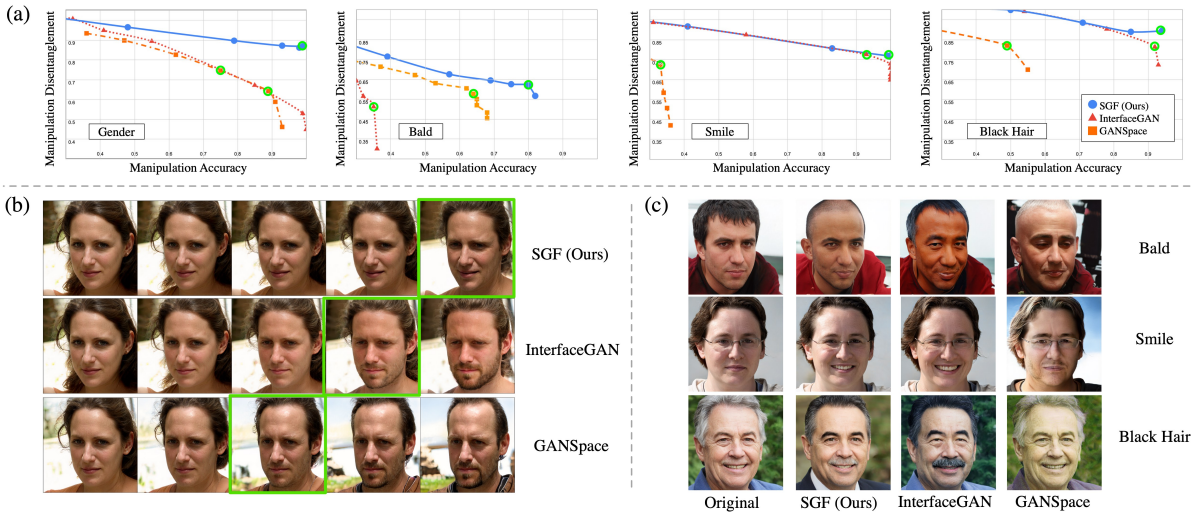

Figure 5. **Comparison of facial attribute editing in the FFHQ-Attributes.** (a) The MDCs of methods for each attribute. The point highlighted with a green circle has highest harmonic mean of accuracy and disentanglement along the curve. (b) "gender" manipulation results of different methods. Green boxes mark the results that use the highlighted hyper-parameters. (c) Manipulation of other attributes. We use the highlighted hyper-parameters of each method.

These results also align with the visual perception for each image in Figure 5(b) and (c). The overall score shows our method can generally achieve better disentanglement with high manipulation accuracy than InterfaceGAN and GANSpace. As GANSpace shows inferior overall performance, we only compare our method with InterfaceGAN in the following experiments.

In our user study for comparison of SGF with InterfaceGAN, 61% of the total queries (244 queries of the total 400 queries) judge our method has a higher degree of disentanglement. Combining the results with the experiments on MDS, we conclude that our method is able to edit attributes with less entanglement compared with other methods.

4.6. Comparison on CelebAHQ-Attributes

The MDS of CelebAHQ-Attributes data are shown in Table 1. Despite using a different GAN model, our SGF still outperforms InterfaceGAN with a similar margin in each attribute in FFHQ data. These results indicate that our method can be applied to different GAN models while maintaining similar performance gains compared to InterfaceGAN.

4.7. Manipulation on Anime-KeypointsAttr

Extending the control conditions to keypoints-attributes, we demonstrate that SGF can use keypoints and attributes to jointly control anime faces. Figure 6(a) shows the sequential editing results of head poses and facial attributes. Our model edit images in a stable and disentangled manner throughout the manipulation process of both keypoints and attributes.

By fine-tuning each facial keypoint, we can add precise facial expression control to anime characters. As shown in Figure 6(b), moving the eyebrows changes the overall expression from natural to sad in the second column. In other columns, we controls the mouth and eyes to change the character's expressions (e.g., angry or happy).

4.8. Manipulation on Flowers-Caption

To further explore the potential of multi-dimensional control, we use natural language as control conditions with the help of sentence embedding. Figure 7 (a) shows that our method can manipulate the color and the shape of generated flowers according to the given target captions.

Figure 7(b) shows manipulation results with different caption compositions. The first row compares the results using captions with similar meanings. While "large and red" and "red and large" produce completely different flowers, both results match the target caption. The images in the second row show the results of color mixing. The manipulation result of "red and purple" is a flower with purplish-red petals. From caption compositions experiments, we suggest that our method can leverage the power of sentence embedding to manipulate latent codes.

4.9. Limitations and Discussions

Some limitations exists for SGF despite the compelling experimental results. Figure 8 shows typical failure cases of SGF. To begin with, SGF does not cover the case where target condition is out of the training data distribution. For an anime image generator trained on aligned face images, faces with unaligned keypoints are out of the generation scope. Therefore, for the results of head yaw modification using keypoints in Anime-KeypointsAttr dataset (Figure 8(a)), the edited faces do not exactly match the given

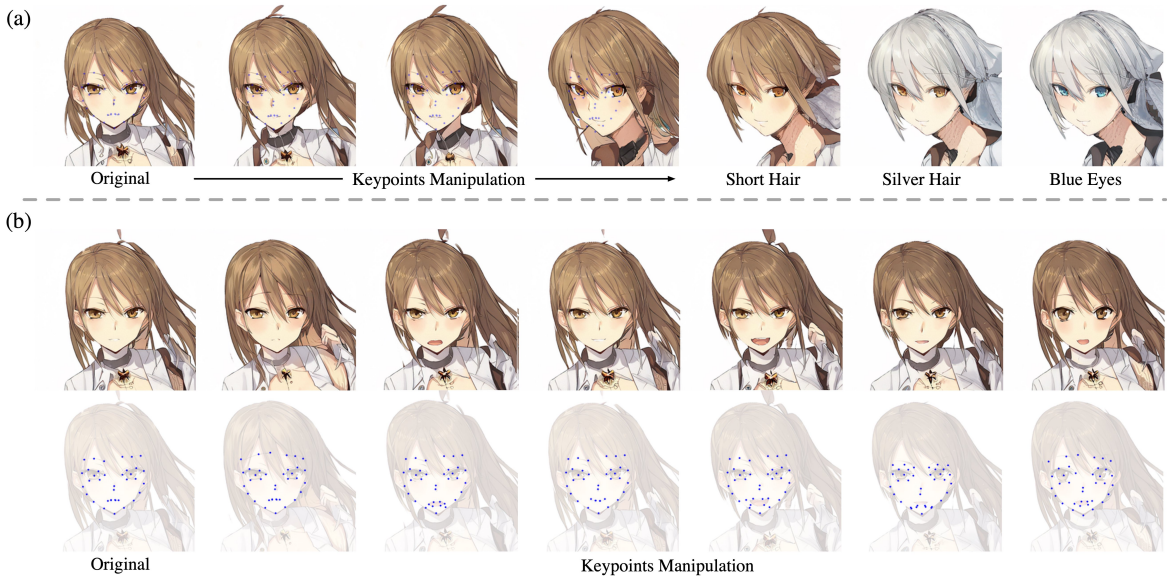

Figure 6. **Manipulation in Anime-KeypointsAttr dataset.** (a) Sequential editing by keypoints (column 1 to 4) and attributes (columns 5 to 7). Target keypoints are shown as blue dots. (b) Keypoints manipulation for expression control, the second row shows the corresponding target keypoint conditions.

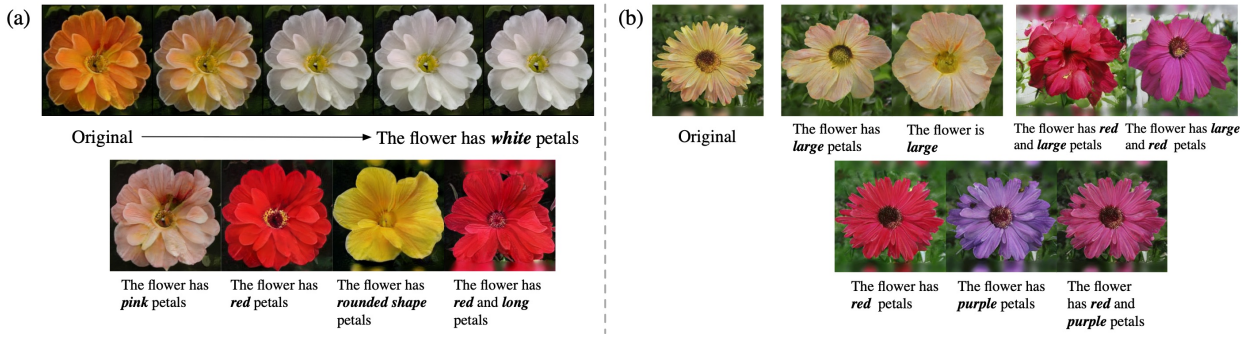

Figure 7. **Manipulation by caption in Flowers-Caption dataset.** (a) Latent space manipulation results on Flowers-Caption using different target captions. (b) Manipulation of Flowers with different caption compositions.

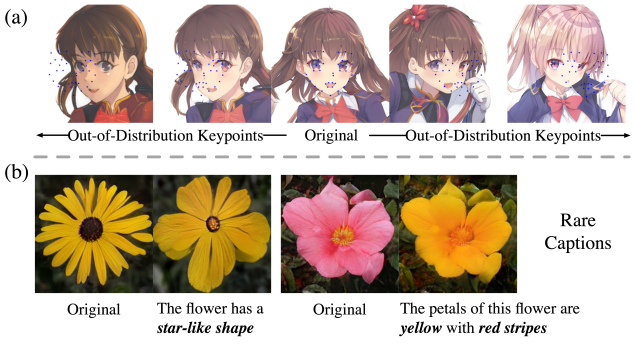

Figure 8. **Typical failure cases of our method.** Refer to Section 4.9 for details.

target keypoints. If the target condition is relatively near the generation scope, our method tends to stop at a point with a similar condition. However, an extremely out-of-distribution target condition may lead to side effects including style and color changes (the leftmost and rightmost images in Figure 8(a)). In addition, there are cases where our model fails to capture conditions that rarely appear. For example, SGF failed to edit flowers in Figure 8(b) because both captions are uncommon in the training dataset. We suggest that building a high-quality dataset with diversity and balanced distribution of condition may be the key to overcome the above limitations.

5. Conclusions

We proposed a unified approach for latent space manipulation on various condition modalities, showed a higher degree of disentanglement in facial attributes editing and able to use facial landmarks as well as natural languages to edit an image. The multi-dimensions control has the potential application to a wide variety of settings and we hope this method will provide interesting avenues for future work.

Acknowledgments. We thank Yingtao Tian for helpful discussions and all reviewers for valuable comments.

References

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

[2] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv preprint arXiv:2008.02401*, 2020.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of ICML*, 2017.

[4] 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):59, 2019.

[5] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *Proceedings of ICML*, 2019.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *Proceedings of ICLR*, 2018.

[7] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Proceedings of IEEE International Conference on Automatic Face & Gesture Recognition*, 2018.

[8] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of CVPR*, 2019.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of NIPS*, 2014.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Proceedings of NIPS*, 2017.

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

[12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of CVPR*, 2018.

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

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of CVPR*, 2017.

[16] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *Proceedings of ICLR*, 2019.

[17] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017.

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

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of CVPR*, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of CVPR*, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

[22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of CVPR*, 2017.

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of ICCV*, 2015.

[24] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of ICLR*, 2018.

[25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[26] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *Proceedings of ICLR*, 2019.

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[28] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of ICML*, 2016.

[29] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*, 2019.

[30] Masaki Saito and Yusuke Matsui. Illustration2vec: a semantic vector representation of illustrations. In *SIGGRAPH Asia 2015 Technical Briefs*. 2015.

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

[32] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *Proceedings of ICML*, 2020.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*, 2015.

[34] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of ICCV*, 2016.

[35] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Proceedings of NIPS*, 2020.

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of ICCV*, 2017.