Latent Transformations via NeuralODEs for GAN-based Image Editing

Valentin Khrulkov[1*] **Leyla Mirvakhabova**[2*]
Ivan Oseledets[2] **Artem Babenko**[1,3]

Yandex[1]
Skolkovo Institute of Science and Technology (Skoltech)[2]
National Research University Higher School of Economics[3]
khrulkov.v@gmail.com, {leyla.mirvakhabova,i.oseledets}@skoltech.ru, artem.babenko@phystech.edu

Abstract

Recent advances in high-fidelity semantic image editing heavily rely on the presumably disentangled latent spaces of the state-of-the-art generative models, such as StyleGAN. Specifically, recent works show that it is possible to achieve decent controllability of attributes in face images via linear shifts along with latent directions. Several recent methods address the discovery of such directions, implicitly assuming that the state-of-the-art GANs learn the latent spaces with inherently linearly separable attribute distributions and semantic vector arithmetic properties.

In our work, we show that nonlinear latent code manipulations realized as flows of a trainable Neural ODE are beneficial for many practical non-face image domains with more complex non-textured factors of variation. In particular, we investigate a large number of datasets with known attributes and demonstrate that certain attribute manipulations are challenging to obtain with linear shifts only.

1. Introduction

Generative Adversarial Networks (GANs) [13] have significantly advanced techniques for image processing and controllable generation, such as semantic image-to-image translation [15, 9, 22, 34, 35] and image editing via manipulating the internal GAN activations [5, 10] or generator parameters [4, 8]. Moreover, since the GAN latent spaces are known to possess semantically meaningful vector space arithmetic, a plethora of recent works explore these spaces to discover the interpretable directions [27, 29, 11, 16, 26, 31, 14, 25]. The directions identified by these methods are then used to manipulate user-specified image attributes, which is shown to be particularly successful for face images [29].

While a large number of methods exploring the latent spaces of pretrained GANs have recently been developed, most of them learn linear latent controls, and more complex nonlinear latent transformations are hardly addressed. We conjecture that this limitation could arise because most of the latent editing literature is biased to the human face datasets, where linear transformations are sufficient for decent editing quality [29].

In this work, we demonstrate that in the general case, the linear latent shifts cannot be used universally for all domains and attributes, and more complex nonlinear transformations are needed. To this end, we analyze how different attribute values are distributed in the latent spaces of GANs trained on several synthetic and real datasets with known attribute labels. Our analysis shows that for non-face images, many attributes cannot be controlled by linear shifts. To mitigate this issue, we propose an alternative parametrization of the latent transformation based on the recent Neural ODE work [7]. Our parametrization allows for gradient-based optimization and can be used within existing methods for latent space exploration [29]. Through extensive experiments, we show that the proposed nonlinear transformations are much more appealing for the purposes of controllable generation. In particular, we show that nonlinear transformations are more beneficial for edits requiring global content changes, such as changing appearance of a scene.

To sum up, our contributions are the following:

- We analyze the distributions of different attribute values in the GAN latent spaces and show that linear latent controls are typically not sufficient beyond the human face domain.
- We propose a Neural ODE-based parametrization of

[*] Equal contribution

the latent transformation that allows for learning the nonlinear controls. On several non-face datasets, we show that usage of this parametrization results in higher editing quality confirmed qualitatively and quantitatively.

- We propose a technique to analyze the learned Neural ODE models and reveal the attributes that require nonlinear latent transformations.

2. Related work

Latent manipulations in GANs. The prior literature has empirically shown that the GAN latent spaces are endowed with human-interpretable vector space arithmetic [27, 29, 11, 16, 31, 36, 30]. E.g., in GANs trained on face images, their latent spaces possess linear directions corresponding to adding smiles, glasses, and gender swap [27, 29]. Since such interpretable directions provide a straightforward route to powerful image editing, their discovery currently receives much research attention. A line of recent works [11, 29] employs explicit human-provided supervision to identify interpretable directions in the latent space. For instance, [29] use the classifiers pretrained on the CelebA dataset [23] to predict certain face attributes. These classifiers are then used to produce pseudo-labels for the generated images and their latent codes. Based on these pseudo-labels, the separating hyperplane is constructed in the latent space, and a normal to this hyperplane becomes a direction, controlling the corresponding attribute. Another work [11] solves the optimization problem in the latent space, maximizing the score of the pretrained model, which predicts the image aesthetic appeal. The result of this optimization is the direction that makes images more aesthetically pleasing. Two self-supervised works [16, 26] seek the vectors in the latent space that correspond to simple image augmentations such as zooming or translation. Finally, a bunch of recent methods [31, 14, 25] identify interpretable directions without any form of (self-)supervision. [31] learns a set of directions that can be easily distinguished by a separate classification model based on two samples, produced from the original latent codes and shifted along the particular direction. [25] learns the directions by minimizing the sum of squared off-diagonal terms of the generator Hessian matrix. Another approach, [14], demonstrates that interpretable directions often correspond to the principal components of the activations from hidden layers of generator networks.

Nonlinear editing. While some works [16, 3, 36] briefly mention the possibility of non-linear latent transformations, they do not provide reliable evidence of the necessity of non-linear editing; therefore, most of the recent editing literature employs only linear manipulations. To the best of our knowledge, our work is the first that demonstrates several cases of inadequacy caused by linear editing and provides a rigorous quantitative comparison with non-linear techniques on several datasets.

3. GAN-based image editing

In this section, we remind on current approaches to controllable image generation and editing via GANs and discuss their possible weaknesses.

We assume that we are given a well-trained GAN model $G : \mathcal{W} \to \mathcal{X}$, where $\mathcal{W} \subset \mathbb{R}^d$ denotes the latent space and $\mathcal{X} \subset \mathbb{R}^{C \times H \times W}$ is the image space. We work with the style–based generators where the manipulation is performed in the so-called *style space* $\mathcal{W}$, which has been shown to be more "disentangled" with respect to various image attributes. We focus on the supervised setting and assume that we are given a trained semantic attribute regressor network $\mathcal{R} : \mathcal{X} \to \mathcal{S} \subset \mathbb{R}^N$, which predicts the attribute values for a given image. Here $\mathcal{S}$ denotes the semantic attribute space of the image domain $\mathcal{X}$, *e.g.*, for human faces, this can be hair color, age, *etc*. The space of image attributes $\mathcal{S}$ may be exhaustive, *i.e.*, a point $\mathbf{x} \in \mathcal{X}$ may be uniquely determined by its attributes $\mathcal{R}(\mathbf{x})$, or be only a subset of a "true" attribute space.

3.1. Manipulating image attributes by shifts

Most of the current approaches propose to manipulate attributes of synthesized images with simple *linear* translations in the latent space. This means the following. Let $\mathbf{s}$ denote the attribute vector of a generated image $G(\mathbf{w}_0)$, and let $\mathbf{s}_i$ be a single chosen (binary) attribute. These approaches seek to carefully construct a vector $\mathbf{n}_i$ such that by gradually changing $\mathbf{w}_0$ as

$$\mathbf{w}(\alpha) = \mathbf{w}_0 + \alpha \mathbf{n}_i, \tag{1}$$

we achieve controls over the value of the attribute $\mathbf{s}_i$. Note that these approaches assume that we use a *single* vector $\mathbf{n}_i$ for all the points $\mathbf{w} \in \mathcal{W}$. InterFaceGAN [29] is among the most successful approaches to construct the shift vector that manipulates a desired attribute in the supervised setting. The idea of this method is to find a *hyperplane* in the latent space separating $\mathbf{w}$ with different values of $\mathbf{s}_i$. For a large number of random style vectors $\mathbf{w}$ the labels are obtained by evaluating $\mathcal{R}[G(\mathbf{w})]$, and the hyperplane is found by fitting an SVM on this synthetic labeled dataset. The corresponding direction $\mathbf{n}_i$ is then simply a normal vector to this hyperplane.

4. Nonlinear approach

In contrast to previously described methods, in this section, we focus on the nonlinear approach to manipulations in GAN latent spaces. We can view the simple linear shift

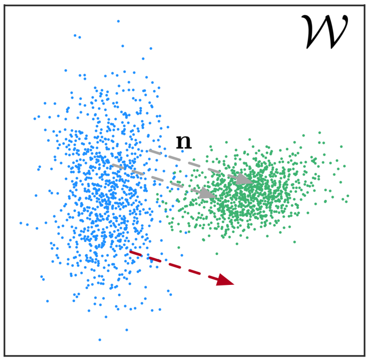


Figure 1. Consider the following toy example of a distribution of a binary attribute $\mathbf{s}_i$ in the latent space $\mathcal{W}$. With any translation vector $\mathbf{n}$, certain points in the left distribution will "miss" the right distribution. This suggests that more complex, *i.e*., nonlinear translations may be necessary.

given by (1) as the flow of a differential equation with a constant righthand side, *i.e*., $\dot{\mathbf{w}} = \mathbf{n}_i$ with the initial condition $\mathbf{w}(0) = \mathbf{w}_0$. The generalization of such edits to the nonlinear domain can be made straightforwardly: *e.g*., by replacing the righthand side with some function, depending on the input $\mathbf{w}$. We propose a simple approach: we consider the Neural ODE model [7] with a righthand side parametrized by a neural network consisting of a few linear layers with the Leaky ReLU activation. The model is later trained end-to-end using the regressor $\mathcal{R}$. Let us now discuss the model structure and training procedure in detail.

4.1. Reminder on Neural ODEs

The Neural ODE model [7] bridges the differential equations and neural networks by parametrizing a system of ODEs:

$$\dot{\mathbf{h}}(t) = f(\mathbf{h}(t), t; \theta), \tag{2}$$

where $t \in [0, T]$ is time and $\mathbf{h}(t) \in \mathbb{R}^d$. The solution of the ODE problem at time step $t = T$ serves as the output of the corresponding hidden layer, where the input is provided as the initial value to (2). In practice, the output can be computed via black-box differential equation solvers. In order to compute gradients with respect to θ, it is common to use the adjoint method, which allows for memory-efficient backpropagation at the cost of extra function evaluations.

4.2. Neural ODE for image manipulation.

We directly apply Neural ODEs for image manipulation performed in the latent space of GANs. *I.e*., we replace the linear flow (1) with the curved flow of a trainable Neural ODE in the latent space. Let us now briefly describe the specifics and the optimization goal.

Network architecture. The righthand side of Neural ODE model $f(\cdot; \theta)$ is represented by a simple multilayer perceptron (MLP) with Leaky ReLU nonlinearity (with $\alpha = 0.2$). We vary the number of layers in the network f from 1 to 3; we additionally consider a constant righthand side, *i.e*., equation of the form $\dot{\mathbf{w}} = \theta$ with trainable θ. We additionally normalize the righthand side of the ODE to the unit length, so the trajectories for all the approaches have the same length for the same value of T. To sum up, our Neural ODE takes the following form.

$$\dot{\mathbf{w}} = \frac{f(\mathbf{w}; \theta)}{\|f(\mathbf{w}; \theta)\|}, \tag{3}$$

with $f(\cdot; \theta)$ being an MLP (or constant) as described above. To compute image edits, we then move along the trajectories of this ODE in the latent space.

Loss function. As was discussed above, we search for transformations in the latent space in such a manner that they would change the desired attribute while leaving the others unchanged. Recall, that $\mathcal{R}$ is a network, which predicts the value of image attributes. Suppose that we have N discrete attributes and our goal is to manipulate the i-th (binary) attribute. Let $\mathbf{w}$ be a random style vector with a vector of attributes $\mathcal{R}(\mathbf{w}) = (\mathbf{s}_1, \ldots, \mathbf{s}_i, \ldots \mathbf{s}_N)$. We set the target attribute vector $\hat{\mathbf{s}} = (\mathbf{s}_1, \ldots, 1 - \mathbf{s}_i, \ldots \mathbf{s}_N)$. After following along the trajectory of the Neural ODE starting at $\mathbf{w}$ for some time value T, we obtain a point $\mathbf{w}(T, \theta)$. More concretely, in practice we set the maximal value $T_{\max}$ of order $8-12$ and then randomly sample the interval $[{}^{T_{\max}}/_4, T_{\max}]$ to get the final time step (as was done in [31]). In what follows, as a slight abuse of notation, we will denote by $\mathcal{R}[\cdot]$ the predicted attribute values of a generated image: $\mathcal{R}[G(\cdot)]$.

To achieve the aforementioned desired transformation properties, we introduce a loss function consisting of two terms: the first one, denoted by $\mathcal{L}_1$, measures the discrepancy between the obtained and the desired i-th attribute values.

$$\mathcal{L}_1(\mathbf{w}, \theta) = \mathrm{CE}(\mathcal{R}[\mathbf{w}(T, \theta)]_i, \hat{\mathbf{s}}_i), \tag{4}$$

where CE stands for the cross entropy loss. The second one represented by $\mathcal{L}_2$ is a loss term controlling the change of remaining attribute values.

$$\mathcal{L}_2(\mathbf{w}, \theta) = \frac{1}{N-1} \sum_{j=1, i \neq j}^{N} \mathrm{CE}(\mathcal{R}[\mathbf{w}(T, \theta)]_j, \hat{\mathbf{s}}_j). \tag{5}$$

Finally, the loss function takes the form $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$. Note that this loss function, in general, can not be written as a single cross-entropy loss since the discrete attributes $\hat{\mathbf{s}}_j$ may belong to spaces of different cardinalities (e.g., we may have an attribute like 'object position' assuming a large set of intermediate values). In our work, we search for a separate Neural ODE for each attribute; however, in principle, it is possible to consider conditional Neural ODEs and have a single model.

4.3. Evaluation

Estimation of the quality of the image editing approach is a nontrivial task, often relying on mean opinion scores provided by human evaluators or some proxy metrics. Typically, given a method to manipulate the latent code of an image, we continuously steer it while visually observing whether the desired attribute shift happened and how disentangled the transformation looks [29, 31]. In a nutshell, we propose to measure these effects numerically in the spirit of traditional PR or ROC curves. Concretely, given a starting point $\mathbf{w}$ and a timestep τ, we measure (i) whether the value of the attribute $\mathbf{s}_i$ is equal to the desired target value and (ii) for each of the remaining attributes we compute the normalized *entropy* of the label distribution along the trajectory up to τ. The idea behind (ii) is that in the ideal case, the attribute values remain constant, and relatively "rare" and localized spontaneous attribute changes are still satisfactory. Formally, for a given $\mathbf{w}$ with the attribute vector $\mathcal{R}(\mathbf{w}) = (\mathbf{s}_1, \dots, \mathbf{s}_i, \dots \mathbf{s}_N)$ and the target attribute vector $\hat{\mathbf{s}} = (\mathbf{s}_1, \dots, 1 - \mathbf{s}_i, \dots \mathbf{s}_N)$ these two metrics, termed $\mathrm{C}(\tau, \mathbf{w})$ and $\mathrm{D}(\tau, \mathbf{w})$ for **C**ontrol and **D**isentanglement, are defined as follows.

$$\mathrm{C}(\tau, \mathbf{w}) = \begin{cases} 1, \hat{\mathbf{s}}_i = \mathcal{R}[\mathbf{w}(\tau)]_i \\ 0, \text{otherwise} \end{cases}, \tag{6}$$

$$\mathrm{D}(\tau, \mathbf{w}) = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \frac{\mathcal{H}\Big(\{\mathcal{R}[\mathbf{w}(t)]_j\}_{t=0}^{\tau}\Big)}{\mathcal{H}\Big(\mathrm{Uniform}(\#\mathcal{S}_j)\Big)}. \tag{7}$$

Here $\#\mathcal{S}_j$ denotes the cardinality of j-th attribute, and $\mathcal{H}$ is the entropy. To get the global values of these metrics, we simply average them across a large number of samples. I.e, we obtain a curve $(\mathrm{C}(\tau) = {}^1/_N \sum_i C(\tau, \mathbf{w}_i), \mathrm{D}(\tau) = {}^1/_N \sum_i D(\tau, \mathbf{w}_i))$, which by construction lies in the unit square. By comparing relative positions of these curves for two methods, we can judge which provides more disentanglement/better control quality. Note that, however, the reliable estimation of the quality by these metrics is only possible in the case when all the factors of variation in data are known, which is possible for synthetic datasets. For large-scale datasets of real images, we have to resort to standard visual evaluation with human assessors.

5. Experiments

We have implemented the proposed method in `Pytorch`. For GAN training, we utilized a single DGX-1 station with 8 Nvidia Tesla V100 GPUs, and for training Neural ODEs, we used a single V100 GPU (in our setting, it takes roughly 30 minutes per single direction). Specifics of architectures, optimization details, and additional experiments, are available in the supplementary material. Our code and models are available at github.

5.1. Synthetic datasets

The goal of this section is to quantitatively verify the benefits of nonlinear editing on several large-scale datasets where the ground truth factors of variations are known and contain both texture and non-texture-based attributes (see Figure 3 for an example).

Datasets.

- *MPI3D* consisting of $1,036,800$ images with 7 factors [12]. This dataset represents a robotic arm in various positions holding an object of varied shapes and colors. We use the toy part of the dataset, *i.e.*, simply rendered images.
- *Isaac3D* is a recently proposed dataset of high-resolution images [24] containing $737,280$ images with 9 factors of variations; we resize images to 128×128 resolution. This is, in a way, an advanced version of *MPI3D* with photorealistic images and more attributes.

For both of these datasets, each image is *uniquely determined* by the corresponding attributes; thus, we can reasonably compare linear and nonlinear manipulations using the aforementioned metric.

GAN Model. We use the recently proposed StyleGAN 2 [19] and its implementation in `Pytorch` found at github. We use the default settings except for the number of layers in the style network, which we set to 3, as was done in [24, 20]. For *MPI3D* we trained the model for 12.5M frames and for *Isaac3D* for 25M frames. We do not use any data augmentation for training.

Attribute regressors. For each dataset, we train an attribute regressor network on the real data. For *MPI3D*, we use a simple four-block CNN as a backbone, followed by multiple classification heads, and for *Isaac3D* we use ResNet18 (not pretrained) as a backbone. In both cases, the attribute regressors were able to achieve more than 99% accuracy on all the attributes on the test set.

Neural ODE models. We consider two Neural ODEs, namely one with a trainable constant righthand side (termed **Ours(linear)** on the plots), and righthand sides represented by MLPs of depth one (**Ours(nonlinear)**). We use the open-source implementation of Neural ODE found at github. We train all the models for 5000 iterations with a batch size of 24. For these two datasets, all the attributes take more than two values, while previously, we considered binary attributes. To alleviate this, when rectifying the attribute with index j for the sake of simplicity, we binarize it

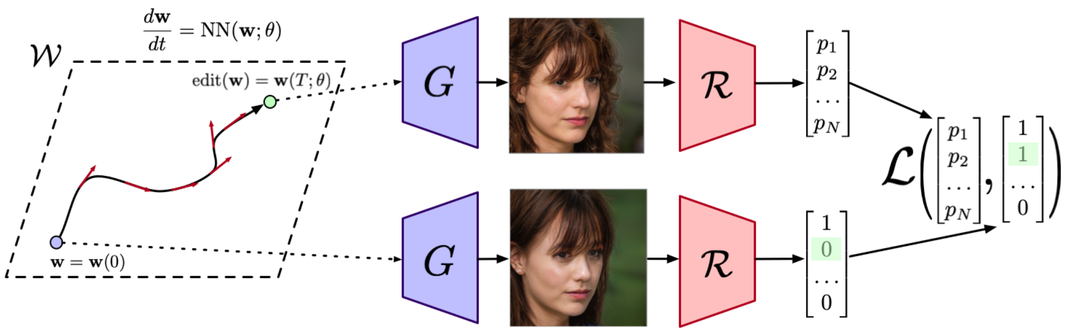
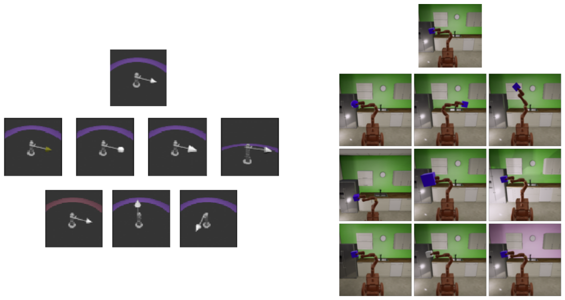

$\frac{d\mathbf{w}}{dt} = \mathrm{NN}(\mathbf{w};\theta)$
$\mathcal{W}$
$\mathrm{edit}(\mathbf{w}) = \mathbf{w}(T;\theta)$
$\mathbf{w} = \mathbf{w}(0)$
G
$\mathcal{R}$
p_1 p_2 … p_N
$\mathcal{L}$
1 1 … 0
1 0 … 0

Figure 2. Our learning method. A sample from the latent space is transformed via the nonlinear flow of a trainable Neural ODE. The loss function ensures that the desired semantic attribute of the edited image changed while others remained fixed. The attributes are obtained using the pre-trained attribute regressor $\mathcal{R}$.



(a) MPI3D (b) Isaac3D

Figure 3. Samples from two synthetic datasets used for quantitative evaluation of the methods. In both cases, all the factors of variation are known.

by learning to transform $\mathbf{s}_j = 0$ to $\mathbf{s}_j = \#\mathcal{S}_j - 1$, and all other attributes retain their full discrete set of values. For both datasets we used $T_{\max} = 12$. For *Isaac3D*, we consider all the attributes, and for *MPI3D*, we train Neural ODE models for the first five attributes due to the large cardinality of the two positional attributes; however, we still include them when computing metrics and during training. As a reference, we include the scores obtained by the InterFaceGAN (**IF**) method and its 'disentangled' version, termed **IF projected**. The latter was obtained using the conditional manipulation approach specified in [29] (we conditioned each attribute on all the remaining attributes).

5.1.1 Evaluation of the learned manipulations

We now evaluate the obtained Neural ODEs. Results for *Isaac3D* are summarized on Figure 4. Here we plot CD-curves as was discussed in Section 4. Intuitively, a lower position of one curve, well-covering the Control range, with respect to another, indicates better quality of the image editing method. We observe that deep Neural ODEs can obtain a reasonable trade-off between disentanglement and control for all the attributes. On the other hand, linear controls are inferior in terms of either control (i.e., they do not work for a subset of the latent space) or provide inferior disentanglement. Interestingly, in some cases, the curves make a jump near the origin, *e.g.*, for `camera_height`. Such behavior indicates that the latent codes have to travel a considerable distance before the attribute shift occurs, which intuitively corresponds to neat well–separated attribute distributions. On the other hand, for many other non-textured attributes, such distributions may "interlace" the latent space $\mathcal{W}$ and the attribute transition can occur relatively close to the point of origin. An example of learned manipulations is provided at Figure 5.

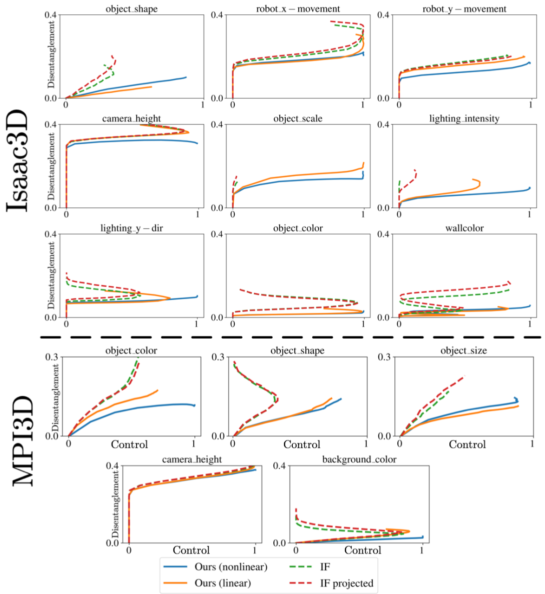


Figure 4. Control-Disentanglement curves for *Isaac3D* and *MPI3D*. We observe that, unlike linear shifts, nonlinear flows allow for achieving good control for all the samples while maintaining reasonable disentanglement.

5.2. Real-life datasets

In this section, we investigate the behavior of nonlinear image edits learned by our Neural ODE-based approach on real-life datasets. Additionally, we include experiments on the *CUB-200-2011* dataset in Appendix B.1.

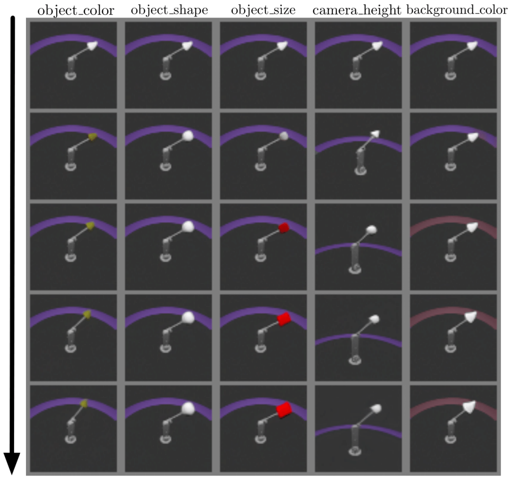


Figure 5. Manipulating the first five attributes on *MPI3D*. For the visualization, we used learned Neural ODEs of depth 2.

Datasets.

- *FFHQ* is a dataset consisting of $70,000$ high-quality human faces images [18]. This is a standard benchmark for image editing as it contains a rich variation in age, ethnicity, lighting, and background.
- *Places365* consists of $1,803,460$ training images with $400+$ unique scene categories [33]. We restrict our dataset to the `outdoor natural` scenes and filter out the ones with the attribute `man-made`. The final version contains 48 classes and $239,457$ images.

Model. For these datasets, we also used StyleGAN 2. For *FFHQ*, we used the recent high–quality pretrained model producing images of resolution 256×256 and provided by the authors of [17] at github. *CUB-200-2011* and *Places365* are especially challenging datasets for generative modeling due to the low number of samples and high sample diversity. We utilize the Adaptive Data Augmentation (ADA) strategy [17], which helps to deal with limited size datasets. We use the authors' implementation in `tensorflow` at the same github link. We train both models for 25M frames with the default config; we only change the number of layers in the style network to 8 to be consistent with the *FFHQ* model. For training, we resize all the images to 256×256.

Attribute regressors. For all the attribute regressors, we use the same frozen ResNet18 backbone pretrained on ImageNet, followed by a trainable MLP of depth two, where for each attribute, we consider a separate classification head. To train the regressors, we utilized the following data.

- For *FFHQ*, we used the data and attribute annotation provided by the CelebA [23] dataset. There are $202,599$ images with 40 binary attributes such as smile, gender, hairstyle, *etc*.
- For *Places365* we used the Transient Attributes [21] dataset. This dataset contains $8,571$ scene images with annotations for 40 binary attributes such as "fog", "snow", "dusk", "autumn". We used all 40 attributes when training the regressor.

Another approach to enforce identity preservation in our method is to utilize an off-the-shelf representation network $\mathcal{F}$, such as FaceNet [28] for human faces datasets. In this case, we replace our $\mathcal{L}_2$ loss with the cosine distance between the representations $\mathcal{F}[G(\mathbf{w}(T;\theta)]$ and $\mathcal{F}[G(\mathbf{w})]$. See supplementary material for the details on this experiment.

Neural ODE. We use exactly the same settings and loss function as for the synthetic datasets. We train a separate model for each attribute. We experimented with Neural ODEs of depth 1 and 2, but we did not notice any significant visual difference, so we chose to stick with depth 1 in our visualizations. We denote it by $\mathbf{Ours}(\mathbf{nonlinear})$. To verify the actual benefit of *nonlinearity* over simply having a more powerful loss function, we also study Neural ODEs with a trainable constant righthand side. We denote it by $\mathbf{Ours}(\mathbf{linear})$.

Baselines. As the baseline approach for supervised image editing, we consider InterFaceGAN (IF). We use $20,000$ latent codes to train SVMs. We did not obtain competitive results with the conditional IF, thus we utilize the standard version. This approach is similar to other works [3, 36].

5.2.1 FFHQ

We hypothesize that for datasets consisting of human face images, the attributes describing *texture*-based features (*e.g.*, hair or skin color) can be manipulated linearly relatively well, while for *non-texture*-based attributes (*e.g.*, hair type, gender), the linear shifts may have slightly worse performance. To support our hypothesis, we experiment with `gender` and `wavy hair`; our findings are described in Figure 6. Additionally, we experiment with a *composition* of attribute manipulations: *e.g.*, we may want to change the gender at first and then manipulate the hair type; our experiments are summarized in Figure 7. We note that in all the experiments, our nonlinear method outperforms or is on par with linear methods in terms of visual quality.

5.2.2 Places365

Similar to the previous reasoning, we consider the attributes which intuitively correspond to the drastic change of image content. Namely, we study the `rugged` attribute and the `lush vegetation` attribute. Results are provided at Figure 8. Here we can observe an interesting failure mode of linear methods: for instance, in the last example, they

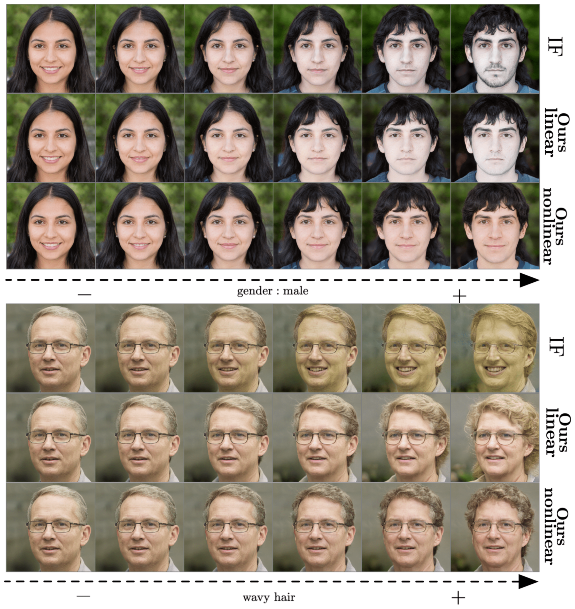


Figure 6. Manipulating two different attributes on the StyleGAN 2 trained on the *FFHQ* dataset. For `gender:male` and `wavy hair` linear shifts suffer from (i) unnatural face color (ii) identity change.

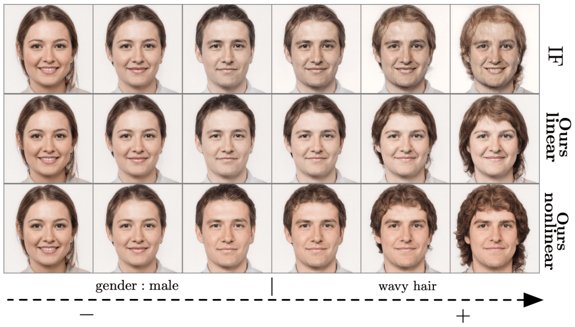


Figure 7. An example of a sequential attribute manipulation: `gender:male` combined with the subsequent `wavy hair` attribute. Our nonlinear method performs visually better with respect to both control and disentanglement.

simply make the texture greener, which on a very high level, corresponds to more "vegetation". However, they struggle to add details like trees or grass, which is successfully achieved by our nonlinear method. Similar results hold for the `rugged` attribute as well.

5.2.3 Editing real images

In this section, we demonstrate that the obtained Neural ODE-based edits can be applied to real images projected to the StyleGAN 2 $\mathcal{W}+$ space. We used the standard projector [19] and trained model of depth 1. As commonly done for real image editing [1, 2, 3] we apply edits to a subset of indices of $\mathcal{W}+$. Concretely, we used the indices (0-6) for this experiment. Our results are provided at Figure 9.

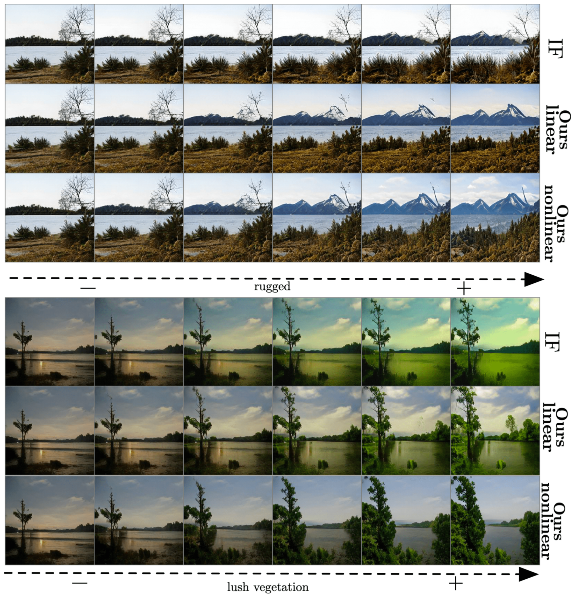


Figure 8. Manipulating `rugged` and `lush vegetation` attributes on the *Places365* dataset. We observe that our nonlinear method achieves the desired control over image contents, while linear shifts tend to change images' texture.

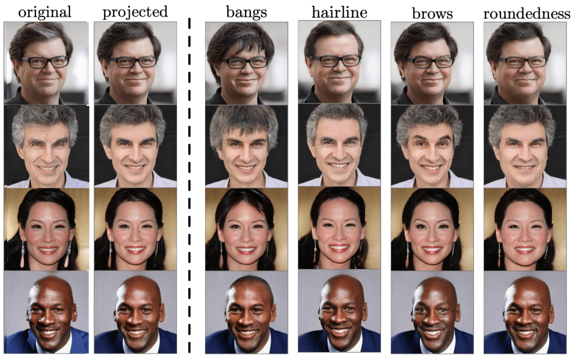


Figure 9. Manipulation of real images embedded in the $\mathcal{W}+$ space.

5.2.4 Human evaluation

Similar to previous works [3], we perform a human evaluation of the quality of the obtained edits. We selected 13 common attributes for *FFHQ* and 32 attributes for *Places365*. During the evaluation, we have presented three images to an assessor: an original image and two modified images obtained by two different methods; these two images were shown in random order. We asked the following questions: (**Q1**)'Which has better attribute change to target <`attr`>?' and (**Q2**)'Which better preserved identity of the original image?'. The possible answers included *Left*, *Right* and *None / both / not applicable*; the total number of participants were 21 and the number of responses was ~ 1000 for both datasets. We compare $\mathbf{Ours}(\mathbf{nonlinear})$ against $\mathbf{Ours}(\mathbf{linear})$ and **IF** methods in separate studies. Results are given in Table 1; we observe that our nonlinear method allows for better control and identity preservation,

	FFHQ		*Places365*	
vs	**IF**	$\mathbf{Ours(linear)}$	**IF**	$\mathbf{Ours(linear)}$
Q1	$+34\%$	$+10\%$	$+47\%$	$+48\%$
Q2	$+4\%$	$+5\%$	-20%	$+31\%$

Table 1. Improvement of $\mathbf{Ours(nonlinear)}$ against linear methods (in absolute percentage values) according to human evaluators.

especially on the more challenging *Places 365* dataset. We noticed that on this dataset **IF** often struggles to make any visual edits, which explains its superiority for **Q2**. We visualize the interface of our questionnaire on Figure 14 in Appendix C. The breakdown of the improvements for each particular attribute (for the $\mathbf{Ours(nonlinear)}$ vs **IF** evaluation) is given in Figure 15 in Appendix C; we also noticed that the top 4 most challenging attributes coincided for **IF** and $\mathbf{Ours(linear)}$, which indicates the need for nonlinearity for certain attributes (`wavy_hair`, `gray_hair`).

5.2.5 Analysis of learned Neural ODEs

In this section, we study the Neural ODEs obtained for various attributes with our method. We focus on the model of depth 1, i.e, it takes the form $\frac{d\mathbf{w}}{dt} = A\mathbf{w} + b$. For the analysis, we ignore the normalization of the right-hand side since it does not affect the obtained trajectories and corresponds only to their reparameterization. To study the obtained ODE, it is convenient to switch to the eigenbasis of A. In these coordinates (assuming all the eigenvalues are real), the ODE takes the simple form $\frac{d\widetilde{\mathbf{w}}}{dt} = \mathrm{diag}(\lambda_1, \dots \lambda_N)\widetilde{\mathbf{w}} + \widetilde{b}$. The eigenvalues of large magnitude $|\lambda_i| \gg 1$ correspond to a 'fast' subspace where some nontrivial dynamics happens. On the other hand, in the 'slow' subspace with $|\lambda_i| \ll 1$, the dynamics is close to linear, i.e., the trajectories are close to straight lines. Thus, we can measure the 'complexity' of an attribute by evaluating how quickly the eigenvalues of the corresponding matrix decay. If they decay rapidly, then this attribute is easier to control with linear shifts and requires more 'nonlinear' controls in the opposite case. One way to estimate how many vectors span the range of the matrix A is via *singular entropy*. It is defined in terms of singular values $\{\sigma_i\}$ of the matrix A in the following way:

$$\mathcal{H}_{SVD}(A) = -\sum_{i=1}^{N} \tilde{\sigma}_i \log \tilde{\sigma}_i, \tag{8}$$

with $\{\tilde{\sigma}_i\}$ being the set of normalized singular values: $\tilde{\sigma}_i = {\sigma_i}/{\sum \sigma_i}$. The values of $\mathcal{H}_{SVD}(A)$ can serve as a proxy to the log-dimensionality of the 'fast' subspace of A. We hypothesize that for the attributes with larger values of $\mathcal{H}_{SVD}$, our nonlinear method provides a more significant improvement. When computing $\mathcal{H}_{SVD}$, we utilize the first 128 singular values (out of 512) in order to get rid of the noise induced by the training procedure (the results are not sensitive to this parameter). The obtained values are provided at Figure 10. To verify our hypothesis, we compute the Spearman rank correlation between the attribute ordering provided by $\mathcal{H}_{SVD}$ and human evaluation ordering visualized at Figure 15. The obtained value is ~ 0.41, confirming the existence of a correlation. Interestingly, we find that even such 'simple' attributes such as `gray_hair` still require a nontrivial trajectory in the latent space. The obtained $\mathcal{H}_{SVD}$ values for *Places365* are available in the supplementary material. Overall, based on the experimental results, we argue that attributes requiring a 'global' content change can not be adequately controlled with linear edits. E.g., for `gray_hair` which is naturally entangled with the facial appearance, we do not simply change the hair color but also make the entire face older. Similar logic holds, for instance, for the `lush` attribute on *Places365*. On the other hand, such attributes as `Smiling` or `Bushy_Eyebrows` require relatively small and localized changes, and we observe that the IF method is on par with our nonlinear model.

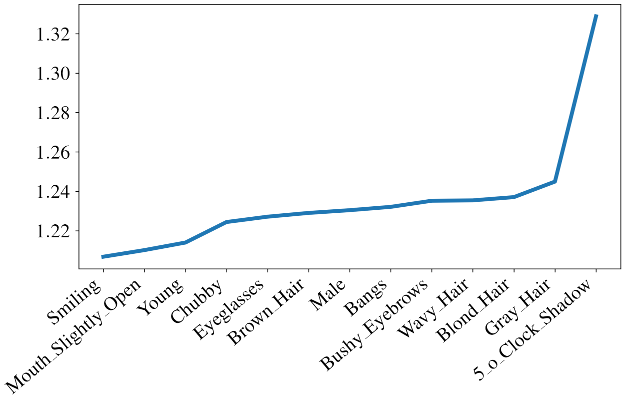


Figure 10. Estimated values of $\mathcal{H}_{SVD}$ for a number of attributes on *FFHQ*.

6. Conclusion

In this work, we discussed a novel approach for image manipulations via nonlinear shifts, parameterized by a Neural ODE model. On multiple datasets, we demonstrated an advantage of our approach over standard linear shifts. For the analysis, we simply considered state-of-the-art StyleGAN 2 trained in a conventional manner. Thus, it may be possible that design choices for this model do not allow for achieving perfect disentanglement. One interesting direction for future work is to better understand the arrangement of attribute distributions in the latent space and how it can be utilized to achieve better disentanglement. Another possible direction to achieve this goal would be to try to tune the GAN architecture so it better incorporates geometrical (*i.e.*, shape) inductive biases.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 7

[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*, pages 8296–8305, 2020. 7

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *TOG*, 2020. 2, 6, 7

[4] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. 2020. 1

[5] 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 *International Conference on Learning Representations*, 2019. 1

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018. 12

[7] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018. 1, 3

[8] Anton Cherepkov, Andrey Voynov, and Artem Babenko. Navigating the gan parameter space for semantic image editing. *arXiv preprint arXiv:2011.13786*, 2020. 1

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 1

[10] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 1

[11] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5744–5753, 2019. 1, 2

[12] Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems*, pages 15740–15751, 2019. 4

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Rrocessing Systems*, 2014. 1

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9841–9850. Curran Associates, Inc., 2020. 1, 2

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[16] Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1, 2

[17] 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, 2020. 6

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 6

[19] 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. 4, 7

[20] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. Disentangled representations from non-disentangled models, 2021. 4

[21] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014. 6

[22] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, pages 1–16, 2020. 1

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 2, 6

[24] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised stylegan for disentanglement learning. In *International Conference on Machine Learning*, pages 7360–7369. PMLR, 2020. 4

[25] William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[26] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuos factors of

variations. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1, 2

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1, 2

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 6

[29] 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*, pages 9243–9252, 2020. 1, 2, 4, 5

[30] Nurit Spingarn, Ron Banner, and Tomer Michaeli. GAN ”steerability” without optimization. In *International Conference on Learning Representations*, 2021. 2

[31] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 1, 2, 3, 4

[32] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 15

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 6

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 1

[35] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017. 1

[36] Peiye Zhuang, Oluwasanmi O Koyejo, and Alex Schwing. Enjoy your editing: Controllable GANs for image editing via latent space navigation. In *International Conference on Learning Representations*, 2021. 2, 6