NEURAL LATENT TRAVERSAL WITH SEMANTIC CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Whilst Generative Adversarial Networks (GANs) generate visually appealing high resolution images, the latent representations (or codes) of these models do not allow controllable changes on the semantic attributes of the generated images. Recent approaches proposed to learn *linear* models to relate the latent codes with the attributes to enable adjustment of the attributes. However, as the latent spaces of GANs are learnt in an unsupervised manner and are semantically entangled, the linear models are not always effective. In this study, we learn multi-stage neural transformations of latent spaces of pre-trained GANs that enable more accurate modeling of the relation between the latent codes and the semantic attributes. To ensure identity preservation of images, we propose a sparsity constraint on the latent space transformations that is guided by the mutual information between the latent and the semantic space. We demonstrate our method on two face datasets (FFHQ and CelebA-HQ) and show that it outperforms current state-of-the-art baselines based on FID score and other numerical metrics.

1 INTRODUCTION

Deep generative models have achieved unprecedented strong performance on generating realistic data particularly in visual domain (Karras et al., 2017; 2019; Brock et al., 2018) by using adversarial losses and variational regularizations. These variational generative models approximate the distribution of true data through a representative and well-regulated latent space where conditions can be set (Mirza & Osindero, 2014) and factors can be transferred across (Karras et al., 2019). Given such flexible latent spaces, it is desirable and imaginable to achieve the capability to control the generation through a human-friendly, continuous scaled and foolproof interface. Such interfaces can be done through linking human-interpretable concepts/attributes with the high-dimensional values of latent codes. Concretely, it requires finding the appropriate latent code corresponding to a desired level of semantic attribute in the image.

This problem has been attempted recently under a simple assumption of the existence of universal, dense linear relations between semantic attributes and latent codes (Shen et al., 2020; Agrawal et al., 2021; Zhuang et al., 2021). However, a recent examination (Locatello et al., 2019) shows that latent spaces learned by these generative models are not properly disentangled according to the semantic attributes that are human understandable. This insight rattles the assumption about simple latent-semantic relations made by the previous works. This also explains the unwanted artifacts in the results of these methods, including affecting untargeted attributes, generating unrealistic images and exaggerating the effects (Härkönen et al., 2020).

In this paper, we re-examine the problem of traversing the latent codes according to a query for a targeted change in the semantic attributes of an image with popular backbone - GAN models. Our analysis suggests that the assumption about single and plain connection across the semantic gap is largely unjustified. In fact, the relations are categorically *non-linear, contextualized, and sparse*. Suggested by this analysis, we design a *neural latent-traversal function* that can model the complex relations between the latent codes and semantic

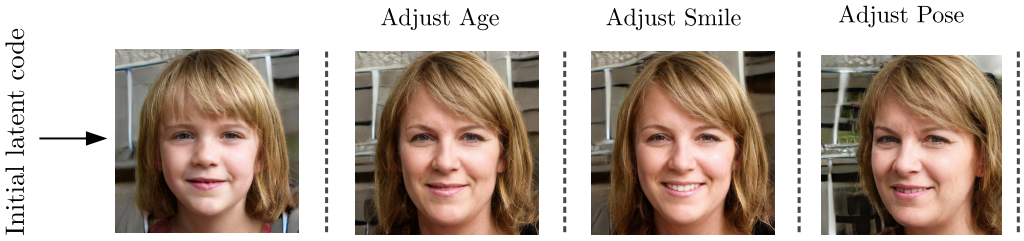


Figure 1: Our approach adjusts the desired attributes of an image, preserving the identity.

attributes. This traversal function supports non-linear relations, can adapt to a detailed context of the traversal, and is constrained according to the sparsity of the relations.

Once efficiently trained from available data, our neural traversal function consistently shows strong performance across benchmark datasets and different backbone GAN models. Especially, this traversal function makes fewer mistakes in untargeted attribute deviation, and is better at controlling exaggerated effects and preventing unrealistic generations as shown in Fig. 1. We call our framework **Se**mantic **N**eural Latent **T**raversal on **Gen**erative models (SeNT-Gen). Our key contributions are:

1. A neural latent traversal method that supports non-linear and contextualized relations between GANs' latent code and semantic attributes;
2. Semantic-tied sparsity constraints to allow component-wise latent traversal based on their relevance to the targeted attribute;
3. Experiments and analysis to verify our hypotheses, and measure the quantitative and qualitative performance of the proposed method compared to state-of-the-art baselines.

2 Re-examining Latent to Semantic Space

We first investigate the relation between the latent space and the semantic space. As the latent space of GAN is learnt in an unsupervised manner, it is semantically entangled, i.e. not semantic-tied. Thus, exploiting this property to *linearly* transform the latent codes to semantic space may not be always justified. To verify this hypothesis, we sampled $50k$ latent codes from the StyleGAN pre-trained model and generated the corresponding images. The Pearson (linear) correlation is then calculated between the 512-dimensional latent codes and the amount of smile extracted by a regressor from the generated images. Fig. 2 (left) shows the majority of latent space dimensions have insignificant (linear) correlation with the amount of smile (presented in a sorted order), i.e. most of the linear correlation values are close to zero. Hence, the amount of smile can be adjusted by a sparse manipulation of latent codes only on the contributing dimensions. Fig. 2 (right) shows the relation between the top-3 latent code dimensions with the amount of smile and confirms that none of them have a strict linear relationship to the amount of smile. These insights form the key motivation for our proposed sparse, non-linear, and contextualized latent traversal method that is capable of modeling a more complex relation between the latent and the attribute space.

2.1 Preliminary

Let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}) \in \mathbb{R}^D$ be a sample of the latent space of a GAN model. Let $\mathcal{G} : \mathbb{R}^D \rightarrow \mathbb{X}$ denote the generator of a trained GAN model, such that $\mathbf{X} = \mathcal{G}(\mathbf{z}) \in \mathbb{X}$, where $\mathbf{X}$ is a sample of the data space $\mathbb{X}$ generated by GAN's generator given latent code $\mathbf{z}$. The data space is defined as $\mathbb{X} \equiv \mathbb{R}^{3\times w\times h}$ in the case of image generation. We define $\mathbf{c} = [c_1, \ldots, c_K]$ as the vector of K continuous attributes of $\mathbf{X}$. We assume that $\mathcal{R}_1(.), \ldots, \mathcal{R}_K(.)$ are K available regressors $\mathcal{R}_k : \mathbb{X} \rightarrow [0, 1],\ k = \{1, \ldots, K\}$ that given an image $\mathbf{X}$ can

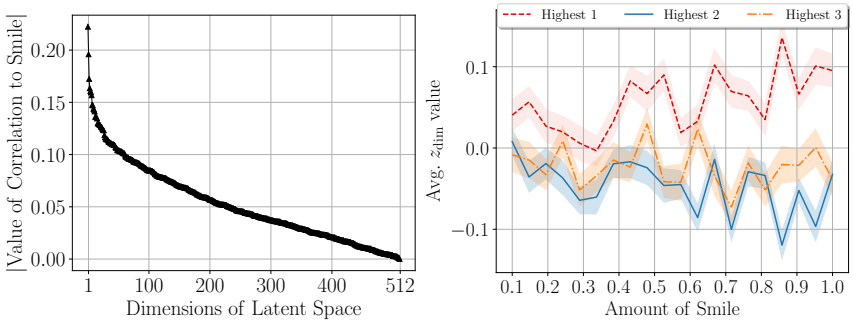


Figure 2: Analysis of latent vs. smile semantic space. (*left*) Most of latent space dimensions do not show a strong (positive/negative) linear correlation to the smile attribute. (*right*) The average values of three highest correlated dimensions of the latent codes with respect to the amount of smile, confirming even highly correlated dimensions do not show a strict linear relation with smile attribute. Shaded region shows one standard deviation.

accurately return the estimated value of the attribute k. These regressors can be human evaluators in an ideal case or accurate pre-trained models in the more practical settings.

2.2 Problem Formulation

The problem is to manipulate the data $\mathbf{X}$ into a modified data $\mathbf{X}^*$ which changes only in the attribute c_k (while keeping other attributes of $\mathbf{X}$ preserved). In the GAN setting, this image will go through an encoder to get the corresponding latent code $\mathbf{z}$ through posterior distribution $q(Z|X)$. We aim to manipulate $\mathbf{z}$ into a modified latent vector $\mathbf{z}^*$ which is defined to be the latent code that generates a solution $\mathbf{X}^* = \mathcal{G}(\mathbf{z}^*)$ with the true attribute $c_k^* = \mathcal{R}_k(\mathcal{G}(\mathbf{z}^*))$. Our proposed solution aims to find this ideal position in the latent space, and arrives at the approximated solution $\hat{\mathbf{z}}$, for which the GAN's generator will result in $\hat{\mathbf{X}} = \mathcal{G}(\hat{\mathbf{z}})$ such that $\mathbf{X}^* \approx \hat{\mathbf{X}}$.

In existing works, the direction and the amount of change is identified by the absolute scale of c_k, defined empirically based on the training dataset. Therefore, users need to find the best alignment of their desired attribute in possible ranges of c_k. This setting is neither intuitive nor consistent. In contrast, we propose a natural, user-intuitive manipulation scheme using relative attribute scale. Given image $\mathbf{X}$, user provides the attribute of interest indexed as k and a corresponding relative change factor r (as a fraction) such that:

$$c_k^* = \max(0, \min(c_k + r * c_k, 1)). \tag{1}$$

Here c_k is defined by an external regressor $\mathcal{R}_k$, hence their scales may change accordingly based on the dataset and the regressor architecture. However the change factor r is consistent and interpretable to the user as long as the domain of c_k is homogenous to linear scaling.

As an example, the user may request the system to increase the smile attribute by 50%, then c_k moves up 1.5 times, i.e. $c_k^* = 1.5c_k$. Accordingly, the model performs the transformation on the latent code to produce a modified latent code $\hat{\mathbf{z}}$ with the desired attribute such that $\mathbb{E}\big[\mathcal{R}_k(\mathcal{G}(\hat{\mathbf{z}}))\big] = c_k^*$. This can be done in a multi-stage manner for all K attributes of the images.

3 LATENT TRAVERSAL NEURAL NETWORKS

3.1 CONTEXTUALIZED LATENT TRAVERSAL

The new formulation of SeNT-Gen allows to implement contextualized traversal functions for each of the attributes. The contextual property is important as it supports sensible adjustment according to the current attributes of the image. As an example, an input image of a young person has more space to traverse to increase the age compared to an image of an elderly person. However, the clamping operation in Eq. 1 may saturate the value of an attribute quickly. To prevent this saturation, we introduce the neural contextualized traversal function $\mathcal{F}_k$ as*:*

$$\hat{\mathbf{z}} = \mathcal{F}_k\big(\mathbf{z}, c_k, c_k^*\big) = \tanh\Big(\big(\sigma(c_k \times \mathbf{w}_c - c_k^* \times \mathbf{w}_c) + \sigma(\mathbf{z} \times \mathbf{w}_\mathbf{z})\big)\mathbf{w}_{\mathbf{z}^*}\Big), \tag{2}$$

where σ is ReLU activation function, $\mathbf{w}_c \in \mathbb{R}^{1\times D}$, $\mathbf{w}_\mathbf{z} \in \mathbb{R}^{1\times 2D}$, and $\mathbf{w}_{\mathbf{z}^*} \in \mathbb{R}^{3D\times D}$.

The architecture of the function in Eq. 2 reflects the intuition that the changes in attribute k triggers an additive modification that is embedded in $\mathbf{z}$. Furthermore, this design saves more parameters than a naive design of MLP neural network and therefore leads to more stable training (see section 5: Ablation study).

This network can be trained with the perceptual loss on the targeted attribute by enforcing the perceived value of c_k of the adjusted image - that is estimated by applying the regressor $\mathcal{R}_k$ on $\mathcal{G}(\hat{\mathbf{z}})$ - to be close to c_k^*:

$$\mathcal{L}_\mathbf{c} = \mathbb{E}_{\mathbf{X}\sim\mathbb{X}}\Big[\big|\big|\mathcal{R}_k\big(\mathcal{G}(\hat{\mathbf{z}})\big) - c_k^*\big|\big|_2^2\Big], \tag{3}$$

where $\hat{\mathbf{z}}$ is the modified version of the sampled latent from posterior distribution $\mathbf{z} \sim q(Z|X)$ defined in Eq. 2. Alternatively, by sampling on the latent space $\mathbf{z} \sim p(Z)$, the loss becomes:

$$\mathcal{L}_\mathbf{c} = \mathbb{E}_{\mathbf{z}\sim\mathbb{Z}}\Big[\big|\big|\mathcal{R}_k\big(\mathcal{G}(\hat{\mathbf{z}})\big) - c_k^*\big|\big|_2^2\Big]. \tag{4}$$

Theoretically, in a well-constructed GAN, the Eq. 3 and 4 are equivalent. However sampling from the latent prior as in Eq. 4 eliminates the use of GAN encoder and therefore reduces moving parts and supports more robust and flexible training.

3.2 SEMANTIC-TIED SPARSE LATENT TRAVERSAL CONSTRAINTS

The traversal function in Eq. 2 is not constrained, therefore in its bare form, it is free to change other attributes together with the intended one. This is a common challenge in related studies on this problem (Shen et al., 2020; Agrawal et al., 2021; Yang et al., 2021). For example, the background or lighting of the face images can change uncontrollably when the user tries to only change facial expression.

To address this issue, we propose to directly constraint the latent traversal using semantic-tied latent traversal constraints. Note that the generic sparsity constraint is not enough as it does not provide the information of which components are semantically relevant to the attributes. To address both of the requirements of sparsity and semantic relevance, we introduce two main constraints. The first pulls $\hat{\mathbf{z}}$ to be close to the correct solution $\mathbf{z}^*$ in the relevant components, while the second constraint pushes its irrelevant components back to their original location $\mathbf{z}$. We implement this design via two loss functions:

$$\mathcal{L}_\mathrm{I} = \mathbb{E}_{\mathbf{z},\mathbf{z}^*\sim\mathbb{Z}}\Big[\big|\big|{\rho_k}^T(\hat{\mathbf{z}} - \mathbf{z}^*)\big|\big|_2^2\Big], \tag{5}$$

$$\mathcal{L}_{\neg \mathrm{I}} = \mathbb{E}_{\mathbf{z},\mathbf{z}^* \sim \mathbb{Z}}\Big[\big|\big|(1-\rho_k)^T(\hat{\mathbf{z}}-\mathbf{z})\big|\big|_2^2\Big], \tag{6}$$

where $\rho_k \in \mathbb{R}^D$ is the thresholded vector of mutual information (MI) values between the dimensions of $\mathbf{z}$ and the c_k values as:

$$\rho_k = \text{threshold}\Big(\text{softmax}\big(\mathrm{MI}(z_i, c_k)\big), \gamma\Big)\ , \forall i = \{1, \ldots, D\}, \tag{7}$$

where γ is a meta-parameter controlling the number of latent components to be modified and thus exploiting sparsity. $\mathrm{MI}(.,.)$ is defined as:

$$\mathrm{MI}(z_i, c_k) = \mathbb{E}\Big[\log \frac{f(z_i, c_k)}{f(z_i)f(c_k)}\Big]. \tag{8}$$

$f(z_i)$ and $f(c_k)$ are density functions of $i-\mathrm{th}$ dimension of the latent space and the $k-\mathrm{th}$ attribute, respectively. $f(z_i, c_k)$ is the joint probability density function of z_i and c_k. As $z_i \sim \mathcal{N}(0, 1)$, Eq. 8 can be estimated by assuming $c_k \sim \mathcal{N}(\bar{c}_k, s^2)$, where $\bar{c}_k$ and s^2 are the sample mean and sample variance of c_k, respectively. The values of mutual information for each dimension of the latent code is then normalized to the range of $[0, 1]$. Following this approach, by applying the thresholding, the dimensions that are highly contributing to a certain attribute are determined.

3.3 Latent Traversal Training

We describe the training data generation for learning our latent traversal function. We sample pairs of latent vectors $(\mathbf{z}, \mathbf{z}^*)$, where $\mathbf{z}$ and $\mathbf{z}^*$ are both sampled from $\mathcal{N}(\mathbf{0}, \mathbb{I})$. The attribute regressors are then used to compute (c_k, c_k^*), where $c_k = \mathcal{R}_k(\mathcal{G}(\mathbf{z}))$ and $c_{k^*} = \mathcal{R}_k(\mathcal{G}(\mathbf{z}^*)),\ k = 1, ..., K$. As the underlying relation of latent and semantic space is unknown, there is no guarantee that the sampled pairs only vary in c_k attribute and not the other attributes of the images. Therefore, we filter out the pairs that have a significant difference on the other attributes. A training pair is defined valid if:

$$\sum_{k' \neq k} \Big|\Big|\mathcal{R}_{k'}\big(\mathcal{G}(\mathbf{z})\big) - \mathcal{R}_{k'}\big(\mathcal{G}(\mathbf{z}^*)\big)\Big|\Big|_2 \leq \epsilon,\ k' = \{1, \ldots, K\},$$

where $\epsilon \geq 0$ is a slack parameter that allows for a maximum amount of difference on other attributes of the paired images.

After generation of the training data, we minimize the objective function of the traversal model $\mathcal{F}_k$ as:

$$\min_\theta\ \mathcal{L}_k \triangleq \lambda_1 \mathcal{L}_{\mathrm{I}} + \lambda_2 \mathcal{L}_{\neg \mathrm{I}} + \lambda_3 \mathcal{L}_{\mathbf{c}}. \tag{9}$$

In summary, based on Eq. 9, $\mathcal{F}_k$ aims to adjust the values of the attributes by indicating the importance of the latent dimensions that have the highest information gain on the changes in an attribute. By doing so, the unrelated dimensions to the targeted attribute will be unmodified, preserving image identity. See further analysis on the objective function in the Ablation study (Section 5) and Appendix 8.1.

4 Experiments

In this section, we first introduce the datasets used in our experiments, we then proceed to our evaluation metrics and the results of our experiments.

Datasets: Two datasets are used in our experiments that are publicly available and also used in prior studies (Shen et al., 2020; Agrawal et al., 2021): (1) Flickr-Faces-HQ Dataset (FFHQ) (Karras et al., 2019) and CelebA-HQ (Karras et al., 2017).

Pre-trained GAN models: We use two pre-trained state-of-the-art GAN models: (1) PGGAN (Karras et al., 2017) and (2) StyleGAN (Karras et al., 2019), both with high-quality face images of 1024×1024 resolution. The latent sample of PGGAN is directly fed into the first convolutional layer of the model. However, StyleGAN provides the option of mapping the latent code from latent space $\mathbb{Z}$ to a more disentangled latent space $\mathbb{W}$ before feeding to the generation in all convolutional layers. We are more interested to analyse the $\mathbb{Z}$ latent space as it is not well-disentangled in contrast to $\mathbb{W}$ latent space with higher level of disentanglement.

Baselines: We compare our experimental results with two recent studies on interpreting GAN's latent spaces: Directional-GAN (Agrawal et al., 2021) and Interface-GAN (Shen et al., 2020). For Interface-GAN, we use their implementation [1]. For Directional-GAN, we followed the implementation details provided by authors.

Regressors: DeepFace (Serengil & Ozpinar, 2020) and Microsoft Face API[2] are used as the pre-trained regressors in our implementations. We select three key facial continuous attributes for analysis in our model: (1) amount of smile, (2) age, and (3) pose. All the values are scaled to $[0, 1]$.

Evaluation benchmark: We propose an extensive numerical metric to evaluate the attribute editing process by alleviating challenges in numerical measurements (Shen et al., 2020). For comparison, for a fixed range of FID scores (Heusel et al., 2017), we adjust a specific attribute then we report the percentage of changes on the targeted attribute and the untargeted attributes, averaged over 5 runs. We consider three ranges of FID scores ($[5, 15], [15, 25], [25, 35]$). The optimal performance will result in the desired percentage of change for the target attribute with minimal changes to the other attributes.

4.1 Quantitative Evaluation

Our quantitative evaluation is a fair comparison between SeNT-Gen and the two related baselines. The objective of this comparison is to confirm if the methods can follow the two main requirements in attribute editing of images: (1) How effectively the methods can change the amount of attribute for a given image? (2) How well can the methods prevent unintended changes of other independent attributes? We use pre-trained regressors to obtain the attribute values after the latent transformations.

Table 1 shows the results of increasing the smile attribute on images that are generated by StyleGAN for three FID ranges. The changes on the attributes are reported in the percentage of change from the initial value of that attribute. For the highest FID score $[25, 35]$, our method is able to increase the targeted attribute the most whilst producing the least change in the untargeted attributes. In the middle range of FID scores $[15, 25]$, our method produces the highest change in the targeted attribute and lower change in one of the untargeted attributes. In the lowest range of FID score $[5, 15]$, Interface-GAN is slightly better on the smile, however, it changes the other untargeted attributes more than our approach. This could be because a linear model may suffice for these lower FID scores. A similar trend can be observed in Table 2 for decreasing the age attribute. Our approach outperforms other baselines in most of the cases as it successfully decreases the age and it also prevents undesired changes to the other independent attributes.

4.2 Qualitative Evaluation

The qualitative analysis aims to provide a visual comparison between SeNT-Gen and the baselines.

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

[2] https://azure.microsoft.com/en-us/services/cognitive-services/face/

Table 1: Results on increasing the amount of smile for SeNT-Gen and baselines on StyleGAN. Higher values for the changes on smile (underlined green column) is better and lower values of undesired changes (shown as absolute values) on other attributes (red columns on changes of age and pose) indicate a more successful preservation of image identity.

Approach	Reference	Deviation on FID	Changes on Smile	Changes on Age	Changes on Pose
Interface-GAN	Increasing Smile	[5,15]	$\mathbf{13.2 \pm 5.0}$	3.8 ± 1.1	3.4 ± 1.3
		[15,25]	15.8 ± 7.8	3.4 ± 2.8	$\mathbf{4.9 \pm 2.4}$
		[25,35]	21.2 ± 8.3	4.2 ± 2.3	11.7 ± 2.9
Directional-GAN		[5,15]	7.3 ± 5.5	3.9 ± 3.3	5.2 ± 2.5
		[15,25]	11.7 ± 7.9	5.2 ± 3.9	11.3 ± 4.4
		[25,35]	19.7 ± 9.1	5.9 ± 4.1	15.8 ± 4.5
SeNT-Gen		[5,15]	12.5 ± 1.6	$\mathbf{2.5 \pm 1.4}$	$\mathbf{3.1 \pm 1.9}$
		[15,25]	$\mathbf{16.3 \pm 4.1}$	$\mathbf{3.0 \pm 1.8}$	5.2 ± 2.8
		[25,35]	$\mathbf{25.1 \pm 5.3}$	$\mathbf{3.4 \pm 2.0}$	$\mathbf{10.4 \pm 2.5}$

Table 2: Results on decreasing the amount of age on StyleGAN. As the age is decreased, lower negative values of changes in the age (underlined green column) is better and lower values of undesired changes (shown as absolute values) on other attributes (red columns on changes of smile and pose) indicate a more successful preservation of image identity.

Approach	Reference	Deviation on FID	Changes on Age	Changes on Smile	Changes on Pose
Interface-GAN	Decreasing Age	[5,15]	$\mathbf{-3.2 \pm 2.1}$	2.3 ± 1.4	1.3 ± 1.5
		[15,25]	2.4 ± 2.9	2.7 ± 1.8	1.3 ± 1.3
		[25,35]	-8.4 ± 3.4	3.9 ± 2.9	$\mathbf{1.6 \pm 1.7}$
Directional-GAN		[5,15]	-2.9 ± 2.6	3.8 ± 1.7	1.5 ± 1.9
		[15,25]	-4.1 ± 3.0	3.9 ± 2.3	1.9 ± 2.7
		[25,35]	-5.5 ± 3.2	5.2 ± 2.8	2.9 ± 3.8
SeNT-Gen		[5,15]	-3.0 ± 1.5	$\mathbf{2.1 \pm 1.2}$	$\mathbf{0.9 \pm 0.7}$
		[15,25]	$\mathbf{-5.9 \pm 2.7}$	$\mathbf{3.0 \pm 1.8}$	$\mathbf{1.0 \pm 1.2}$
		[25,35]	$\mathbf{-9.3 \pm 3.6}$	$\mathbf{3.5 \pm 1.9}$	1.7 ± 1.5

Table 3 details different ranges of attribute values used in the baselines and SeNT-Gen. Both of the baselines ask for an absolute value for the adjustment of an attribute. For a fair comparison, we scale attribute range of Interface-GAN to $[0, 1]$ and report the changes on an attribute in terms of percentage $(\%)$ of increase/decrease from the initial value of that attribute.

Fig. 3 shows the results of increasing the amount of smile on a face image generated by StyleGAN. This example shows that SeNT-Gen outperforms the baselines by showing a better performance in preserving the identity of the image while increasing the amount of smile. Fig. 4 illustrates the adjustment of pose on a sample face image from StyleGAN. Similarly, it can be seen that SeNT-Gen successfully adjusts different values of poses on the face image. Whilst Interface-GAN seems to preserve the identity of the image, it does not achieve all the possible values of pose. Directional-GAN however, successfully adjusts the amount of pose, but cannot preserve the identity of image. Additional results on PGGAN and changing other attributes can be found in Appendix 8.3.

Table 3: Attribute ranges in baselines and SeNT-Gen.

Approach	Range
Interface-GAN	absolute, $[-3, 3]$
Directional-GAN	absolute, $[0, 1]$
SeNT-Gen	relative, $[0, 1]$

(a) 20% (b) 40% (c) 60% (d) 80% (e) 100%

Figure 3: Adjusting the amount of smile by manipulation of a latent sample. Whilst all approaches increase the amount of smile, SeNT-Gen preserves the identity of the image better than others. e.g. The color of the hat and preserving the eyeglasses.

(a) −100% (b) −50% (c) Initial (d) +50% (e) +100%

Figure 4: Adjusting the amount of pose: Whilst SeNT-Gen and Directional-GAN successfully achieve all degrees of pose, Interface-GAN fails at extreme values. It also affects the identity preservation by Directional-GAN.

5 Ablation Study

In this section, we verify the effectiveness of components of our proposed approach: (1) The amount of sparsity which is controlled by thresholding value, (2) the architecture of neural traversal function, and (3) necessity of non-linear mapping of latent to semantic space. We evaluate the variations of SeNT-Gen on a task of increasing the amount of smile. We report the average amount of changes on the attributes for the edited images with the FID score of range $[15, 45]$.

On the Sparsity parameter (γ): Table 4 shows different runs of SeNT-Gen with different values of γ (corresponding to different levels of sparsity). As expected, higher values of γ permit more dimensions of latent samples to be modified, i.e. less sparse manipulation. Hence, as a side-effect of relaxing this constraint, unintended changes on the other attributes of images increase. By removing the semantic-tied sparsity constraint ($\gamma = 1$), the identity preservation of SeNT-Gen fades. Fig. 9 in Appendix shows the visual results of latent sample manipulation with different ranges of γ for a latent code.

On non-linear modeling in neural traversal function: To confirm the importance of non-linearity in SeNT-Gen, all the non-linear activation functions of SeNT-Gen are replaced with the linear activation function, we call this variation *SeNT-Gen-linear*. We also show the benefits of the additive modification module in our neural traversal function model by removing the additive component (see Eq. 2) and replacing it with a dense layer created by concatenating $\mathbf{z}$, c_k and c_k^*. We call this implementation of our model *SeNT-Gen-MLP*.

Table 5 details the results obtained by two variations of SeNT-Gen. It can be seen that our proposed non-linear neural traversal function outperforms other variations of our method by achieving a higher value of smile and lower values of unintended changes on other attributes.

6 Related Work

We only discuss the related works that work with a pre-trained GAN model. These approaches only rely on manipulation of latent codes sampled from a pre-trained GAN model. The existing approaches on discovering the latent code manipulations fall into two categories: (1) *supervised* and (2) *unsupervised*. The supervised approaches generally rely on linear hyperplanes in the latent space to model attributes. InterFaceGAN (Shen et al., 2020) assumes that for any attribute, there exists a hyperplane in the latent space serving as the boundary

Table 4: Ablation result on the sparsity parameter γ. Higher values of changes on smile and lower changes on untargeted attributes are better. The Sparsity column indicates the ratio of dimensions that are thresholded (not selected) to all dimensions.

	The Attributes			
Approach	**Sparsity %**	**Changes on Smile**	**Changes on Age**	**Changes on Pose**
SeNT-Gen ($\gamma = 0.2$)	92.18%	18.1	**3.4**	**5.9**
SeNT-Gen ($\gamma = 0.6$)	80.47%	19.1	5.3	8.4
SeNT-Gen ($\gamma = 1.0$)	0.0%	**24.8**	7.7	9.1

Table 5: Ablation results on (1) using a different architectures of neural network, (2) using a different linear activation function. Higher values of changes on smile and lower changes on untargeted attributes are better.

	The Attributes		
Approach	**Changes on Smile**	**Changes on Age**	**Changes on Pose**
SeNT-Gen-MLP	17.0	8.6	18.4
SeNT-Gen-linear	17.9	7.0	9.1
SeNT-Gen	**18.5**	**3.1**	**6.5**

dividing the latent space into two subspaces regarding presence/absence of an attribute. Directional-GAN (Agrawal et al., 2021) learns linear hyperplanes that models the values of attributes in the latent space. Using these hyperplanes and given values of targeted attributes, Directional-GAN transforms the latent code into a desirable subspace with the specified attributes. Both of these approaches assume linear mapping from latent to (semantic) attribute space.

In contrast, Zhuang et al. (Zhuang et al., 2021) find meaningful linear directions in latent space of GANs to permit image editing by introducing multiple attribute transformations that encode a direction and a magnitude in the latent space. The latent code is then traversed in these directions to adjust the attributes. The use of non-linear mappings from the GAN's generator parameters to semantic attributes has been investigated in GuidedStyle (Hou et al., 2020), but the non-linearity has not been attempted on the latent codes ($\mathbb{Z}$ space) of GANs.

Some very recent studies on unsupervised GAN based image manipulation (Cherepkov et al., 2021; Voynov & Babenko, 2020) discover a set of directions in the space of the generator weights instead of the latent space. The authors report that merely changing the GAN parameters can provide useful image manipulations. Similar to supervised approaches, these studies also aim to leverage the assumption of linear mapping from the latent to the semantic attribute space, but with an unsupervised strategy.

7 Conclusion

In this paper we propose a neural latent traversal method that supports non-linear and contextualized relations between GANs' latent code and semantic attributes. To ensure the identity preservation of images, semantic-tied sparsity constraints are introduced to allow component-wise latent traversal based on their relevance to the targeted attribute. We demonstrate our method on two datasets and showed that our method outperforms existing methods both in quantitative and qualitative measurements.

References

Shradha Agrawal, Shankar Venkitachalam, Dhanya Raghu, and Deepak Pai. Directional gan: A novel conditioning strategy for generative networks. *arXiv preprint arXiv:2105.05712*, 2021.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Anton Cherepkov, Andrey Voynov, and Artem Babenko. Navigating the gan parameter space for semantic image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3671–3680, 2021.

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

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. *Advances in neural information processing systems*, 30, 2017.

Xianxu Hou, Xiaokang Zhang, Linlin Shen, Zhihui Lai, and Jun Wan. Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing. *arXiv preprint arXiv:2012.11856*, 2020.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Sefik Ilkin Serengil and Alper Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 23–27. IEEE, 2020. doi: 10.1109/ASYU50717.2020.9259802.

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*, pp. 9243–9252, 2020.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pp. 9786–9796. PMLR, 2020.

Huiting Yang, Liangyu Chai, Qiang Wen, Shuang Zhao, Zixun Sun, and Shengfeng He. Discovering interpretable latent space directions of gans beyond binary attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12177–12185, 2021.

Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. *arXiv preprint arXiv:2102.01187*, 2021.

Algorithm 1 SeNT-Gen Training Procedure.

Input:
A pre-trained GAN model $\mathcal{G}$ with a noise distribution $\mathbf{z} \sim \mathbb{Z}$.
K attributes and the pre-trained regressors.
Number of training samples N.
Setting loss parameters $\lambda_1, \lambda_2, \lambda_3$ and the thresholding (sparsity) parameter γ.
for $k \in 1, \ldots, K$ **do**
  $cnt = 0$ //Counter of training samples
  while $cnt \leq N$ **do**
    Sample a pair of $(\mathbf{z}, \mathbf{z}^*) \sim \mathbb{Z}$.
    Calculate $\Big(c_k = \mathcal{R}_k\big(\mathcal{G}(\mathbf{z})\big), c_k^* = \mathcal{R}_k\big(\mathcal{G}(\mathbf{z}^*)\big)\Big)$.
    if $\sum_{k' \neq k} \Big|\Big|\mathcal{R}_{k'}\big(\mathcal{G}(\mathbf{z})\big) - \mathcal{R}_{k'}\big(\mathcal{G}(\mathbf{z}^*)\big)\Big|\Big|_2 \leq \epsilon,\ k' = \{1, \ldots, K\}$ **then**
      Accumulate $(\mathbf{z}, c_k, c_k^*)$ to the training input data and $\mathbf{z}^*$ to the training target data. //If sample is valid
      $cnt+ = 1$
    end if
  end while
  Compute $\rho_k = \mathrm{MI}\big(z_{\dim}, c_k\big)$ (Eq. 8) for all dimensions of latent space.
  for Iteration $\in 1, \ldots, M$ **do**
    Calculate $\mathcal{L}_{\mathbf{c}} = \mathbb{E}_{\mathbf{z} \sim \mathbb{Z}}\Big[\big|\big|\mathcal{R}_k\big(\mathcal{G}(\hat{\mathbf{z}})\big) - c_k^*\big|\big|_2^2\Big]$. //Eq. 4
    Calculate $\mathcal{L}_{\mathrm{I}} = \mathbb{E}_{\mathbf{z}, \mathbf{z}^* \sim \mathbb{Z}}\Big[\big|\big|{\rho_k}^T(\hat{\mathbf{z}} - \mathbf{z}^*)\big|\big|_2^2\Big]$.//Eq. 5
    Calculate $\mathcal{L}_{\neg \mathrm{I}} = \mathbb{E}_{\mathbf{z}, \mathbf{z}^* \sim \mathbb{Z}}\Big[\big|\big|(1 - \rho_k)^T(\hat{\mathbf{z}} - \mathbf{z})\big|\big|_2^2\Big]$. //Eq. 6
    Optimize $\min_\theta\ \mathcal{L}_k = \lambda_1 \mathcal{L}_{\mathrm{I}} + \lambda_2 \mathcal{L}_{\neg \mathrm{I}} + \lambda_3 \mathcal{L}_{\mathbf{c}}$. //Minimizing the loss
    Perform gradient descent w.r.t. θ_k.
  end for
end for

8 Appendix

8.1 Training of SeNT-Gen

Algorithm 1 details the training procedure of our proposed approach. Note that the training procedure is defined for all attributes $k = 1, \ldots, K$ in $\mathcal{F}_k$.

8.2 Implementation Detail

8.2.1 Latent Traversal Neural Networks

As mentioned in Ablation studies (see section 5), we implemented two variations of our approach: *(1) SeNT-Gen-linear* and *(2) SeNT-Gen-MLP. SeNT-Gen-linear.*

SeNT-Gen-linear architecture is designed as:

$$\mathcal{F}_k^{linear}\big(\mathbf{z}, c_k, c_k^*\big) = \Big(\big((c_k \times \mathbf{w}_c - c_k^* \times \mathbf{w}_c) + (\mathbf{z} \times \mathbf{w}_{\mathbf{z}})\big)\mathbf{w}_{\mathbf{z}^*}\Big).$$

where $\mathbf{w}_c \in \mathbb{R}^{1 \times 512}$, $\mathbf{w}_{\mathbf{z}} \in \mathbb{R}^{1 \times 1024}$, and $\mathbf{w}_{\mathbf{z}^*} \in \mathbb{R}^{1536 \times 512}$.

SeNT-Gen-MLP architecture is designed as:

$$\mathcal{F}_k^{MLP}(\mathbf{z}, c_k, c_k^*) = \tanh\Big((c_k + c_k^* + \mathbf{z})\mathbf{w}_{\mathbf{z}^*}\Big).$$

where $\mathbf{w}_{\mathbf{z}^*} \in \mathbb{R}^{1536\times512}$.

8.2.2 HYPERPARAMETERS

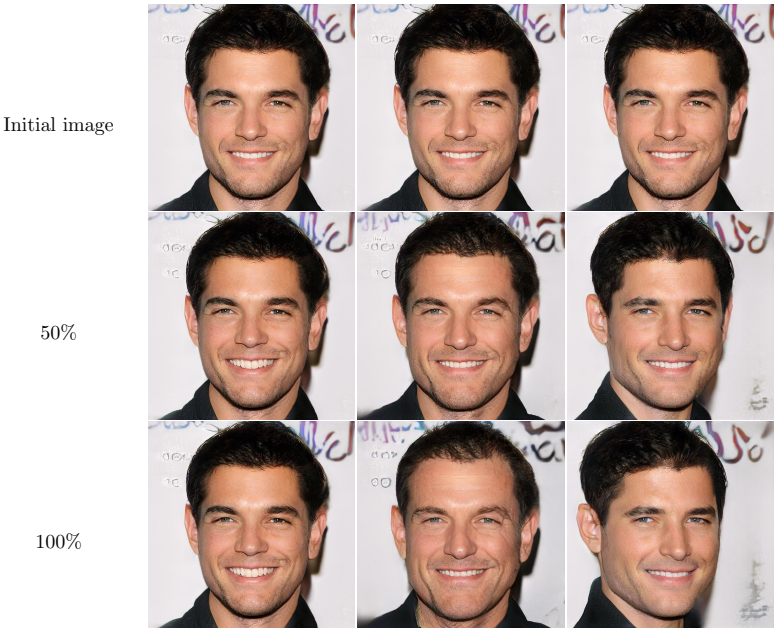
- We set $\lambda_1 = 0.2, \lambda_2 = 0.7, \lambda_3 = 0.1, \epsilon = 0.1$ for all the experiments.
- Thresholding parameter that works as the intensity of sparse constraints is set to $\gamma = 0.25$.
- All the values of latent samples $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ are scaled to $[-1, 1]$.

8.2.3 REGRESSORS

- DeepFace[3] is used to extract the smile (happiness) and the age feature.
- We used Microsoft Face API[4] for obtaining the pose values of images.

8.3 ADDITIONAL RESULTS ON PGGAN AND STYLEGAN

Additional visual results are shown in the following figures.



(a) Adjusting smile (b) Adjusting age (c) Adjusting pose

Figure 5: An example of adjusting the attributes of images generated in PGGAN by our approach.

[3]https://github.com/serengil/deepface
[4]https://azure.microsoft.com/en-us/services/cognitive-services/face/

(a) -100% (b) -50% (c) Initial (d) $+50\%$ (e) $+100\%$

Figure 6: Adjusting the amount of smile by manipulation of a latent samples (StyleGAN).

(a) -100% (b) -50% (c) Initial (d) $+50\%$ (e) $+100\%$

Figure 7: Adjusting the amount of age by manipulation of a latent samples (StyleGAN).

(a) -100% (b) -50% (c) Initial (d) $+50\%$ (e) $+100\%$

Figure 8: Adjusting the amount of pose by manipulation of a latent samples (StyleGAN).

8.3.1 THRESHOLDING PARAMETER

Following section 5 in ablation studies, Fig. 9 shows the performance of SeNT-Gen on manipulation of a latent code with different values of γ. When $\gamma \to 1$, the sparsity approaches to zero and all the components of the latent codes will get involved. As expected, the identity preservation of SeNT-Gen will fade in this case.

Figure 9: Ablation study on γ, the threshold factor.