Rayleigh EigenDirections (REDs): Nonlinear GAN latent space traversals for multidimensional features

Guha Balakrishnan[1,2], Raghudeep Gadde[2], Aleix Martinez[2], and Pietro Perona[2]

[1] Rice University, Houston TX 77005, USA
[2] Amazon, Seattle WA 98109, USA

Abstract. We present a method for finding paths in a deep generative model's latent space that can maximally vary one set of image features while holding others constant. Crucially, unlike past traversal approaches, ours can manipulate arbitrary multidimensional features of an image such as facial identity and pixels within a specified region. Our method is principled and conceptually simple: optimal traversal directions are chosen by maximizing differential changes to one feature set such that changes to another set are negligible. We show that this problem is nearly equivalent to one of Rayleigh quotient maximization, and provide a closed-form solution to it based on solving a generalized eigenvalue equation. We use repeated computations of the corresponding optimal directions, which we call Rayleigh EigenDirections (REDs), to generate appropriately curved paths in latent space. We empirically evaluate our method using StyleGAN2 and BigGAN on the following image domains: faces, living rooms and ImageNet. We show that our method is capable of controlling various multidimensional features: face identity, geometric and semantic attributes, spatial frequency bands, pixels within a region, and the appearance and position of an object. Our work suggests that a wealth of opportunities lies in the local analysis of the geometry and semantics of latent spaces.

1 Introduction

Latent spaces of deep generative networks like generative adversarial networks (GANs) [13, 17, 18, 29] and variational autoencoders (VAEs) [19] are known to organize semantic attributes into disentangled subspaces without supervision [14, 16, 29, 37, 39]. This property is the basis of several latent space *traversal* algorithms that can modify specific image attributes while holding others constant by moving along carefully-chosen latent space directions [4, 12, 28, 31, 43]. Traversal methods have many potential applications including dataset creation/augmentation, image editing, entertainment and graphic design.

Virtually all existing traversal methods assume *scalar* attributes of interest that may be modeled well with global linear functions, e.g., a linear regressor

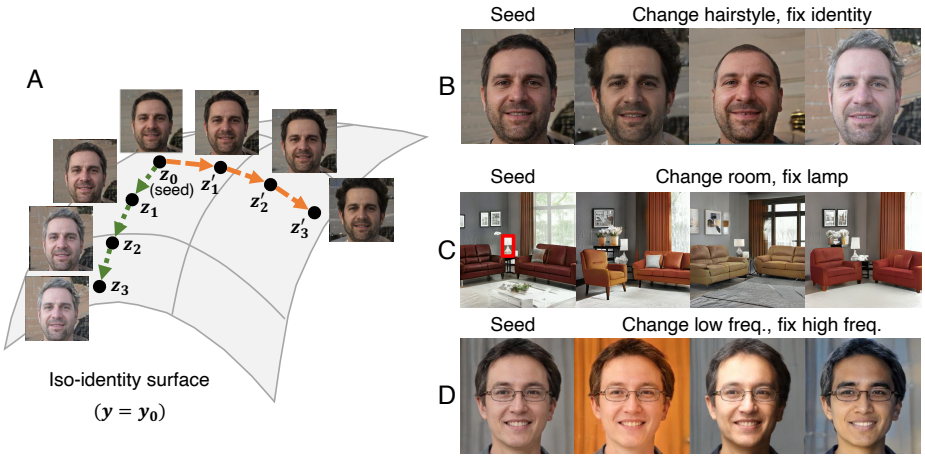

Fig. 1. Method and examples. (A) Our method traverses the local latent space around a seed point $\mathbf{z}_0$ along optimally chosen paths to synthesize images that share the same high-dimensional attribute value $\mathbf{y}_0$ (e.g., identity), and vary as much as possible across other image attributes (e.g., hairstyle). Also shown are samples from our method modifying (B) hairstyle while preserving identity (C) a living room with a fixed object (lamp in red box of seed image) and (D) low spatial frequencies while modifying high ones. These results are produced by our method using StyleGAN2 generators.

or a support vector machine, in the latent space. This approach works well for attributes like gender, hair color and smile of faces [4, 31] and image transformations like translation, color change and camera movements [16, 28]. But these approaches cannot be easily extended to work with attributes like 'style of a couch' and 'face identity' which are best described with high-dimensional vectors.[3] For example, to find a latent space traversal that preserves identity in our experiments, we need a representation that can compute the similarity between two 512-dimensional embeddings returned by a face recognition model [10]. In addition, faces with the same identity or rooms with the same furniture layout (see Fig. 1C) tend to be tightly clustered in latent space, requiring methods tuned to local latent space geometry unlike the common global linear models used for scalar attributes.

We propose a method to tackle this broader class of traversal problems. Given a point in latent space, we aim to generate many traversals, or sequences of images, such that we vary one multidimensional feature ($\mathbf{x}$) in as many ways as possible subject to other multidimensional features ($\mathbf{y}$) being held approximately constant. We formalize the task of finding local latent directions that

[3] There is no physical 'identity' ground truth behind a GAN-generated portrait. However, human observers or face recognition algorithms can respond to the question "Is this the same person?" and can produce consistent judgments. Therefore 'identity' here denotes 'perceptual identity'.

fulfill these criteria as a constrained optimization problem. By using differential approximations of the feature functions, we recast the problem into an instance of Rayleigh quotient maximization, which has a well-known closed-form solution (Sec. 3.1). The principal directions that solve this problem, which we call Rayleigh EigenDirections(REDs), span the local latent subspace containing good paths. Using REDs, we propose a fast linear and more accurate iterative nonlinear projection traversal algorithms (Sec. 3.3) to produce arbitrary-length paths. Our approach is agnostic to network architecture, scene content, and choice of attribute embedding functions.

We evaluate our method using StyleGAN2 [17, 18] and BigGAN [7] generators. We consider a number of challenging applications outside the scope of previous GAN traversal algorithms: face traversals that preserve identity (Fig. 3) while changing hairstyle and facial geometries, face traversals that preserve/change content from specific spatial frequency bands (Fig. 5), and living room traversals that preserve the appearance and location of selected pieces of furniture (Fig. 6). We provide a number of qualitative results demonstrating the perceptual quality of our generated image sequences, and quantitatively demonstrate the necessity for nonlinear traversal strategies in these applications. Finally, we also compare our method against well-known global linear model baselines [4, 31] for scalar attributes and perform comparably, though with some failure cases that we discuss in Sec. 5.

Our main contributions are: (a) REDs, a *local* method for synthesizing a diverse set of images that share a chosen set of multidimensional attribute. The method is principled, simple, and versatile – applicable to pretrained generators, any image type, and to both low-level and semantically meaningful features. (b) A nonlinear technique for long-distance traversals in latent space; (c) Qualitative and quantitative validation experiments on a number of challenging synthesis tasks in different image domains (faces, livingrooms and ImageNet) using two different models (StyleGAN2 and BigGAN).

2 Related Work

Several studies focus on finding interpretable directions in GAN latent spaces for editing and synthesizing images. Most propose finding global linear directions correlated with scalar attributes of interest [4, 14, 12, 28, 31, 38, 43]. Unfortunately, multidimensional features like face identity and image regions lie on complex latent space manifolds rather than on simple linear ones, meaning that a single global direction is not appropriate to model them. Recently, a method called LowRankGAN [46] was proposed for manipulating image regions by finding low-rank subspaces around a latent point. As in our work, they compute a local Jacobian matrix to discover steerable latent subspaces that change one feature while fixing another. Our work is similar in spirit to LowRankGAN with the following differences: we cast the traversal problem as a constrained optimization related to a generalized eigenvalue problem, we propose and demonstrate the superiority of nonlinear traversals for multidimensional features (see *Projection*

algorithm in Sec. 3.3), and our method is applicable to general multidimensional features. Another work [41] introduces a new disentangled intermediate latent space where linear traversals along a single dimension provides control over specific visual properties by computing local gradients. Overall, our nonlinear traversals produce samples that satisfy the enforced constraints for longer traversal lengths compared to [41] and [46].

A few nonlinear traversal strategies for scalar features also exist. Several are based on training deep neural networks to map latent codes to features [16, 36, 44]. Our method is complementary to these – ours requires no additional training, but also does not leverage global latent space structure as theirs presumably can. Finally, our focus on a local rather than global view of the latent space may also complement various theoretical studies on understanding GAN latent space structure [3, 5, 8, 21, 30, 39].

A more explicit way to control GAN outputs is to train the generator using attribute values as inputs. Many of these so-called "conditional GANs" have been proposed, particularly for altering face attributes [2, 6, 9, 15, 20, 22–26, 34, 35, 42, 45], controlling face identity [6, 32–34], and conditioning on semantic maps [27, 40]. Our approach is complementary to all of these in that offers the benefit of not needing to design and train a GAN from scratch with apriori-known attribute controls. Working with a general-purpose black-box GAN has the advantage of keeping all control objectives open and not committing to a specific goal, e.g. preserving identity, from the beginning.

3 Method

Given a point $\mathbf{z}_0 \in \mathcal{R}^d$ in latent space defining an image, we want to generate a set of images that holds fixed the multidimensional features $\mathbf{y}_0 \in \mathcal{R}^n$ while maximally changing the features $\mathbf{x}_0 \in \mathcal{R}^m$. For ease of explanation, we assume $\mathbf{y}_0$ and $\mathbf{x}_0$ each define a single multidimensional feature like facial identity or hairstyle, though our method easily handles features from multiple semantic attributes as explained in Sec. 3.2.

We denote the function that computes the *fixed* features $f(\cdot) : \mathbf{z} \rightarrow \mathbf{y} \in \mathcal{R}^n$, and the function that computes the *changing* features $c(\cdot) : \mathbf{z} \rightarrow \mathbf{x} \in \mathcal{R}^m$. For example, in one of our experiments with faces, $f(\cdot)$ is the concatenation of two functions: the GAN generator on the input latent vector, and a face recognition embedding model on the synthesized face. $c(\cdot)$ may be the generator itself (i.e., $\mathbf{x}$ are the raw pixels of the image) or the concatenation of the generator with learning models computing various image attributes.

Starting at $\mathbf{z}_0$, our method traverses different paths in latent space to generate latent code sequences. For each such trajectory t of length L, $\mathbf{z}_0, \mathbf{z}_1^t \cdots, \mathbf{z}_L^t$, we want $\mathbf{y}_i^t \approx \mathbf{y}_0$ for all i and $\mathbf{x}_0, \mathbf{x}_1^t, \cdots, \mathbf{x}_L^t$ to progressively change such that $\|\mathbf{x}_i^t - \mathbf{x}_{i+1}^t\| < \|\mathbf{x}_i^t - \mathbf{x}_{i+2}^t\|$, where $\|\cdot\|$ is a norm. We return all points from all sequences.

The key intuition behind our approach is that there exists a manifold on which $\mathbf{y}$ does not change around $\mathbf{z}_0$ (see Fig. 1). This is true whenever $d > n$

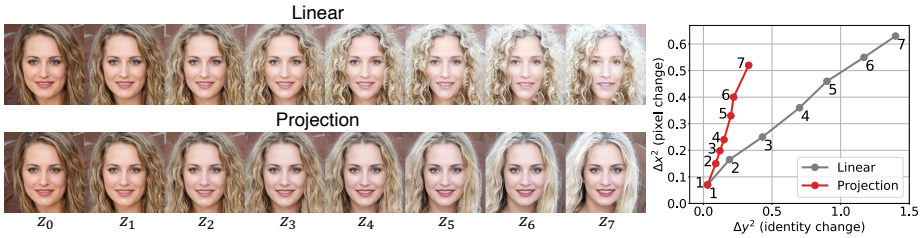

Fig. 2. Comparison of *Linear* and *Projection* traversal. We show a *Linear* and *Projection* traversal originating from the same latent seed code (left-most face), and top RED vector at the seed. $f(\cdot)$ measures identity and $c(\cdot)$ measures raw face pixel values. We also plot squared pixel distance versus squared identity distance. *Projection* and *Linear* change pixels by roughly the same amount, but *Projection* is better at preserving identity (lower distance values).

(and thus the iso-$\mathbf{y}$ manifold has dimension $n-d$) and the generator function is continuous (which, by inspection, it is, apart from a zero-size set). When $d \leq n$, our approach naturally transitions to a "soft" constraint $\mathbf{y}_i \approx \mathbf{y}_0$ as will become clear below. We find directions, which we call Rayleigh EigenDirections (REDs), that maximally change $\mathbf{x}$ within this subspace. This procedure is described in Sec. 3.1. We propose two traversal strategies using REDs in Sec. 3.3: a linear method which simply extrapolates the local REDs throughout the latent space, and a nonlinear method (*Projection*) which updates traversal directions based on local latent space geometry.

3.1 Rayleigh EigenDirections (REDs)

Let $\mathbf{z}$ be a generic point in the generator's latent space with fixed and changing features $\mathbf{y} = f(\mathbf{z})$ and $\mathbf{x} = c(\mathbf{z})$. Given a displacement $\delta\mathbf{z}$, the displacements to $\mathbf{y}$ and $\mathbf{x}$ are:

$$\delta\mathbf{y} = f(\mathbf{z}+\delta\mathbf{z}) - f(\mathbf{z}) \tag{1}$$

$$\delta\mathbf{x} = c(\mathbf{z}+\delta\mathbf{z}) - c(\mathbf{z}). \tag{2}$$

We aim to find the displacement $\delta\mathbf{z}^*$ that maximizes $\delta\mathbf{x}$ with insignificant changes to $\delta\mathbf{y}$:

$$\delta\mathbf{z}^* = \underset{\delta\mathbf{z}:\|\delta\mathbf{z}\|=\epsilon}{\operatorname{argmax}} \|\delta\mathbf{x}(\mathbf{z},\delta\mathbf{z})\|^2 \quad \text{s.t. } \|\delta\mathbf{y}(\mathbf{z},\delta\mathbf{z})\|^2 \approx 0, \tag{3}$$

where we write $\delta\mathbf{x}$ and $\delta\mathbf{y}$ as functions of $\mathbf{z}$ and $\delta\mathbf{z}$, and ϵ is a small, fixed constant. For sufficiently small ϵ, we can approximate $\delta\mathbf{y}$ and $\delta\mathbf{x}$ with local linear expansions: $\delta\mathbf{y} \approx J_f(\mathbf{z})\delta\mathbf{z}$ and $\delta\mathbf{x} \approx J_c(\mathbf{z})\delta\mathbf{z}$, where $J_f \in \mathcal{R}^{n\times d}$ and $J_c \in \mathcal{R}^{m\times d}$ are Jacobian matrices. Letting $A_f(\mathbf{z}) = J_f^T(\mathbf{z})J_f(\mathbf{z})$ and $A_c(\mathbf{z}) = J_c^T(\mathbf{z})J_c(\mathbf{z})$:

$$\delta\mathbf{z}^* = \underset{\delta\mathbf{z}:\|\delta\mathbf{z}\|=\epsilon}{\operatorname{argmax}} \delta\mathbf{z}^T A_c(\mathbf{z})\delta\mathbf{z} \quad \text{s.t. } \delta\mathbf{z}^T A_f(\mathbf{z})\delta\mathbf{z} \approx 0 \tag{4}$$

Algorithm 1: Compute local REDs (solves optimization (4))

Input: A_f, A_c, β_f
Output: R (REDs matrix with directions as columns, from best to worst)
$A_f, A_c \leftarrow A_f/\|A_f\|_2, A_c/\|A_c\|_2$
$\mathbf{u}_f, V_f \leftarrow \text{eig}(A_f)$
$\rho \leftarrow$ smallest k s.t. $\sum_{i=0}^{k} \mathbf{u}_f^2(i) \geq \beta_f \|\mathbf{u}_f\|^2$
$\text{null}(A_f) \leftarrow$ columns ρ to d of V_f
$\tilde{\mathbf{u}}_c, \tilde{V}_c \leftarrow \text{eig}(\text{null}(A_f)^T A_c \ \text{null}(A_f))$
$R \leftarrow \text{null}(A_f)\tilde{V}_c$

This optimization is similar to one of finding the $\delta\mathbf{z}$ that maximizes the Rayleigh quotient $\left(\delta\mathbf{z}^T A_c(\mathbf{z})\delta\mathbf{z}\right) / \left(\delta\mathbf{z}^T A_f(\mathbf{z})\delta\mathbf{z}\right)$, known to be the solution of the generalized eigenvalue problem $A_c\delta\mathbf{x} = \lambda A_f\delta\mathbf{x}$, or the principal eigenvector of $A_f^{-1}A_c$ (see Supplementary). The main point of difference is that in our applications A_f is often singular ($n < d$) and therefore not invertible. Put another way, $f(\cdot)$ is constant in a subspace $\text{null}(A_f)$ around $\mathbf{z}$ and any $\delta\mathbf{z}$ in that subspace will exactly satisfy the constraint in (4). We instead first project A_c onto $\text{null}(A_f)$, and then find the principal eigenvectors of the resulting matrix (Alg. 1) [11]. We return the eigenvectors (REDs) in matrix $R \in \mathcal{R}^{d\times d}$, ordered from best to worst.

For some high-dimensional features, the rank of $\text{null}(A_f)$ may be too small (or even 0 when $d < n$), yielding little to no diversity of $\mathbf{x}$ in the generated trajectories. To address this, we introduce hyperparameter β_f in Alg. 1 that lets users smoothly control the approximation of A_f's rank based on explained variance.

The main computational cost of finding REDs is in calculating the Jacobian matrices J_f and J_c. We compute them using one-sided finite difference approximations with step size ϵ, which requires $d + 1$ forward evaluations of $f(\cdot)$ and $c(\cdot)$. See Sec. 4.6 for more on this topic.

3.2 Handling multiple attributes

To fix multiple attributes $\mathbf{y}^1, \cdots, \mathbf{y}^{n_f}$, we replace the constraint in (4) with multiple constraints: $\delta\mathbf{z}^T A_f^i(\mathbf{z})\delta\mathbf{z} \approx 0, i = 1 \cdots n_f$, and introduce a separate β_f^i for computing the rank of each A_f^i. We compute REDs by projecting A_c onto $\cap_{i=1}^{n_f}\text{null}(A_f^i)$ – the intersection of the fixed attribute nullspaces – and returning the eigenvectors of the resulting matrix as before. To change multiple attributes, we compute REDs separately for each changing attribute, and return all vectors formed by summing together one RED chosen from each set.

3.3 Traversal Algorithms

We propose two traversal algorithms using REDs. The first is a simple *Linear* traversal (see Supplementary for algorithm). We randomly select a direction in the span of R_0 (the REDs of $\mathbf{z}_0$), and generate a sequence of latent codes

$\mathbf{z}_1, \cdots, \mathbf{z}_K$ by moving in that direction starting from $\mathbf{z}_0$ with step size s. In the likely case that the constant-$\mathbf{y}$ manifold is curved, the linear traversal is expected to diverge quadratically from $\|\delta\mathbf{y}\| = 0$ as a function of $\|\delta\mathbf{z}\|$.

Our second algorithm, *Projection* (see Supplementary for algorithm), addresses this shortcoming by recomputing the space of local REDs along the traversal path. We again start by selecting a random direction in R_0. However, at each step i (of length s), we project the previous direction, $\delta\mathbf{z}_{i-1}$, onto R_i. This results in a path that more faithfully adheres to the local geometries of $f(\cdot)$ and $c(\cdot)$ in latent space.

A visual example of a *Linear* and *Projection* traversal for the same initial latent code is shown in Fig. 2, where $f(\cdot)$ measures identity and $c(\cdot)$ measures raw face pixels. *Projection* is better than *Linear* at preserving identity for long trajectories (right plot), while achieving similar levels of image change (left plot).

4 Experiments

We focus our evaluations on two image domains: faces and living rooms, modeled with StyleGAN2 [18]. For faces, we use the public *config-f* model from NVIDIA trained on the Flickr Faces HQ (FFHQ) dataset at 1024×1024 resolution. For living rooms, we train a StyleGAN2 generator from scratch on an in-house dataset of 100K 1024×1024 living room scenes from the web. For both domains, we use the "style" space, $\mathbf{w} \in \mathcal{R}^{512}$, as our latent space. We also demonstrate results on BigGAN [7] trained on ImageNet, where we use the input noise space (in $\mathcal{R}^{128}$) as our latent space.

We set β_f to 0.95 or 0.99 for our experiments, depending on the perceptual characteristics of the generated samples that the user prefers ($\beta_f = 0.95$ results in more diverse samples at the expense of letting the fixed features change to a larger degree). See Supplementary for further analysis and figures.

4.1 Identity, hairstyle and geometry traversals for faces

We first evaluate our method on controlling three multidimensional facial features: identity, hairstyle, and geometry (quantified by 3D facial landmark positions). We use ArcFace [10], a popular open-source face identification model, to encode identity with a 512-dimensional vector. To encode hairstyle, we run a public face segmentation model [4] on each image, set pixels outside of the hair region to 0, and flatten all pixels into a $256\times256\times3 = 196,608$-dimensional vector. We encode 3D geometry using the MediaPipe mesh model [1], which predicts 468 landmarks around the face. This results in a $468 \times 3 = 1404$-dimensional vector. We set β_f to 0.95.

We performed four experiments: changing hairstyle while keeping landmarks fixed, changing hairstyle while keeping landmarks and identity fixed, changing landmarks while keeping hairstyle fixed, and changing landmarks while keeping

[4] https://github.com/zllrunning/face-parsing.PyTorch

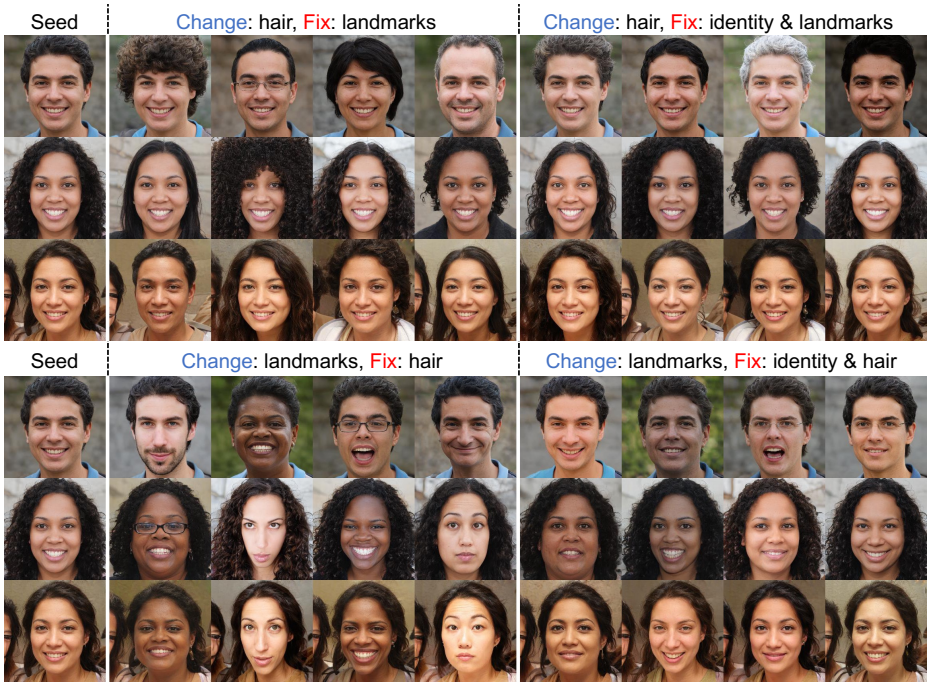

Fig. 3. Results for face traversals controlled by hairstyle, landmarks (geometry) and identity. We generated results using REDs with *Projection*. For each seed face and experiment, we selected four illustrative samples from the first few principal trajectories. (Top) Changing hairstyle while fixing facial landmarks (columns 2-5) and changing hairstyle while fixing identity and facial landmarks (columns 6-9). (Bottom) Changing landmarks while fixing hairstyle (columns 2-5) and changing landmarks while fixing identity and hairstyle (columns 6-9). Our method is able to generate a perceptually diverse set of faces while adhering to the fixed attribute constraints. Explicitly fixing identity greatly helps preserve the identity in the seed images. See Fig. 4 for quantitative analysis.

identity and hairstyle fixed. Fig. 4 presents sample results for three test seed points using REDs and *Projection*. We set both the Jacobian finite difference step and path step s to 1, and the path length $L = 4$. Along with changing the input images along the intended features, our method is able to produce a *wide variety* of different samples from different paths.

We quantitatively evaluated REDs against three baseline direction-finding approaches: choosing directions at random (**Random**), choosing the most significant eigenvectors of A_c, thereby maximizing changes to $\mathbf{x}$ (**Max-$\Delta\mathbf{x}$**), and choosing the least significant eigenvectors of A_f, thereby minimizing changes to $\mathbf{y}$ (**Min-$\Delta\mathbf{y}$**). The plots in Fig. 4 present our results for two of the experiments. When using *Linear* traversal, REDs outperforms the three baseline direction-finding approaches. Max-$\Delta\mathbf{x}$ finds directions that significantly change

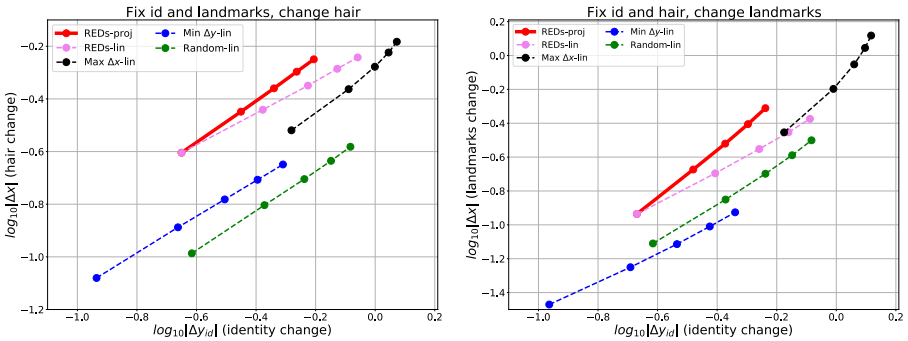

Fig. 4. Quantitative comparison of face traversal strategies controlled by identity, landmarks (geometry), and hairstyle (see Fig. 3 for visual samples). We generated 6 traversals with $L = 4$ steps for each method for 50 random seeds. We plot changes to hair (left) and landmarks (right) versus changes to identity in log-log scale, where each dot in the plot is the average value for each traversal step over all examples. ***Leftward and higher values are better***. Our method using *Linear* traversal (REDs-lin) outperforms the baselines also using *Linear*. Our method with *Projection* traversal (REDs-proj) outperforms REDs-lin by reducing identity changes with no impact to hair or landmark changes.

hairstyle/landmarks and identity, Min-$\Delta\mathbf{y}$ preserves identity but also minimally changes hairstyle/landmarks, and Random performs worst of all. The figure also shows that when using REDs, *Projection* outperforms *Linear*. See Fig. 2 for a visual sample of this comparison and Supplementary for complete traversals and more plots.

4.2 Frequency band traversals

Our method can handle arbitrary low-level image representations. We demonstrate this by controlling specific spatial frequency bands for StyleGAN2 and BigGAN in Fig. 5. We let $f(\cdot)$ and $c(\cdot)$ encode the raw pixels of low-pass and high-pass filtered versions of the input image (and vice versa). We set β_f to 0.99. High-pass modifications change fine details like face physiognomies and expressions, while low-pass modifications mainly change colors, lighting and shading.

4.3 Object-preserving living room traversals

We next apply our method to living room scenes. We aim to keep selected furniture fixed while changing other parts of the scene. We generated furniture bounding boxes with an object detector. We let $f(\cdot)$ encode the raw pixels within the bounding box, and let $c(\cdot)$ encode all remaining pixels in the scene. We set the Jacobian finite difference step to 0.75, path step $s = 0.25$, and a path length $L = 10$. We set β_f to 0.99.

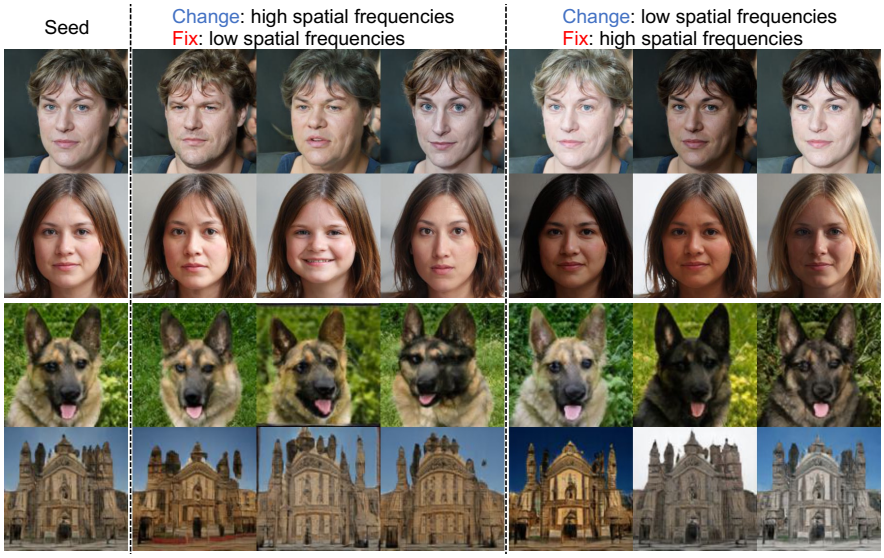

Fig. 5. Samples from traversals controlled by spatial frequency bands. The first two rows are generated by StyleGAN2, and the last two rows are generated by BigGAN. (Columns 2-4) The embedding function $f(\cdot)$ returns the raw pixels of the low-pass filtered image and $c(\cdot)$ the high-pass one. High-pass modifications change fine details. (Columns 5-7) $f(\cdot)$ and $c(\cdot)$ are inverted. Low-pass modifications change colors, lighting and shading.

Fig. 6 shows several sample sequences from REDs with *Projection* traversal and LowRankGAN [46]. Samples from LowRankGAN deviate significantly from the desired constraint of preserving the object in the bounding box, due mostly to the linear traversal strategy used in that method. See caption of Fig. 6 for a detailed description. In Supplementary, we show sample strips of full traversals and also compare against StyleSpace [41]. We observe two notable degradations in all methods the farther we move away from the seed image. First, the 'fixed' object often moves slightly at each step. Second, artifacts become more prominent because we rapidly advance to low-probability regions of the latent space.

4.4 Spatial region traversals for faces

We next demonstrate our method's effectiveness at manipulating facial regions (mouth and eyes). We use the same fixed bounding boxes for the mouth and eyes used in the LowRankGAN study [46]. The changing features are the pixels within the box, and the fixed features are pixels outside the box. Fig. 7 presents sample visual results using REDs with *Projection* traversal. Our method is better than LowRankGAN and StyleSpace [41] at generating a variety of changes within the bounding boxes while roughly adhering to the constraints (the degree to fixing these constraints can be adjusted with β_f).

Fig. 6. Object-preserving living room traversals. We used REDs with *Projection* traversal, with $f(\cdot)$ and $c(\cdot)$ encoding raw pixel values inside and outside a bounding box on a piece of furniture (red box on seed image at left). The object within the box often stays fixed, but can undergo stylistic changes and movements (examples in rows 3, 5) due to feature correlations in latent space. There are diverse changes to the rooms outside of the boxes, including new furniture (rows 2, 3, 5), wall and window properties/decorations (all rows), and house plants (rows 3, 4). Samples in the right three columns are edits following the approach in [46] using the same path step. Clearly REDs-Proj is better at preserving objects.

4.5 Scalar attribute traversals

We finally compare our method against a popular technique for modifying *scalar* attributes with global linear directions [4, 31]: train a linear model (regressor for a continuous attribute or an SVM for a binary attribute) to predict an attribute value from the latent code, and change the attribute by moving along the hyperplane's normal direction. To fix attributes, we orthogonalize the changing attribute's direction with respect to the fixed attribute directions.

Fig. 8 presents our results for four face attributes: age, pose, smile, and gender. Overall, REDS-proj achieves similar qualitative performance to the baseline for most samples, but also has more failures cases when changing an attribute

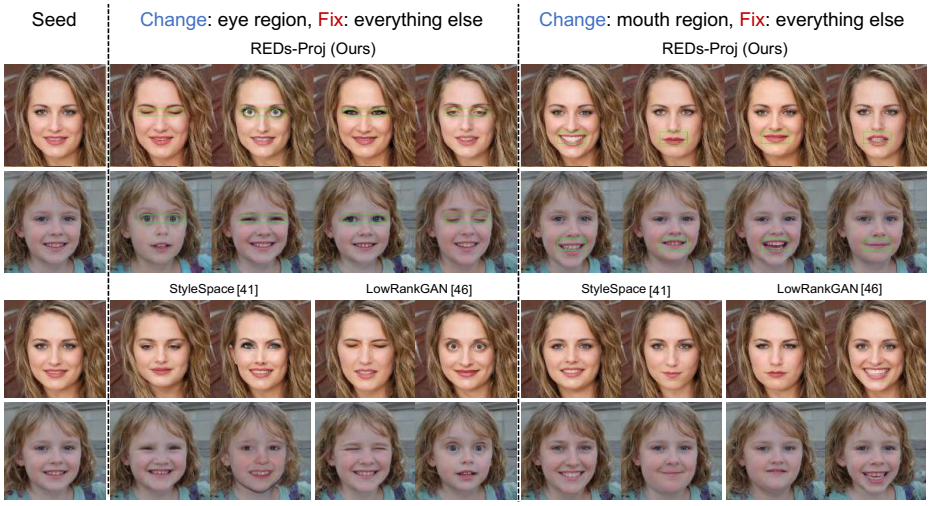

Fig. 7. Samples from traversals controlled by spatial image regions. We let the changing features be the pixels inside a bounding box (green boxes overlaid on images for visualization), and the fixed features be pixels outside the box. (Top) For each seed, we show several output samples of our method changing the eyes (columns 2-5) and mouth (columns 6-9). (Bottom) Results for the same task using StyleSpace [41] and LowRankGAN [46]. Notice the change in other attributes like identity and landmarks which REDS-proj better preserves.

like gender, which often does not have a large local gradient in latent space. We discuss this more in 5.

4.6 Computation Time

Virtually all the computation time of our method is spent on computing the Jacobian matrices in each traversal step, which involves generating $d+1$ images and evaluating $d+1$ feature functions. For the livingroom traversals shown in Sec. 4.3 at 1024x1024 resolution, evaluating one of the d dimensions (assuming no parallelization) on an NVIDIA A100 GPU required approximately 25 milliseconds with StyleGAN2, translating to 15 seconds for all d dimensions. For faces, the time for one Jacobian computation ranged from 15-50 seconds depending on the features being extracted. This time may be reduced dramatically if operations are parallelized in batches and across multiple GPUs.

5 Discussion

Our experiments demonstrate the effectiveness of REDs at finding locally optimal orientations. By contrast, selecting random traversal directions or local directions that prioritize only one of the objective or constraint in Eq. (4) do not work well due to the high dimensionality of the latent space (see Fig. 4).

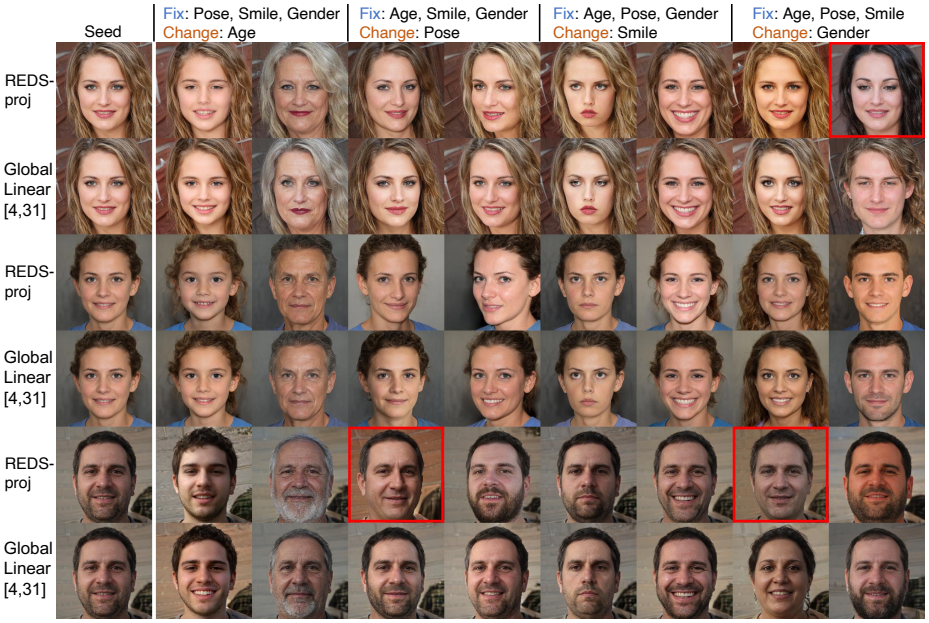

Fig. 8. Samples from traversals controlled by scalar semantic attributes. On scalar attributes one may compare our method to a baseline of using a global linear model (SVM or ridge regressor) in latent space [4, 31] (the global method is not defined and cannot handle multi-dimensional attributes). We change one attribute (age, pose, smile, gender) at a time while fixing the other three. Both methods are comparable for many cases. REDs sometimes fails (red-boxed images), particularly for gender (see Sec. 5 for further discussion).

The superiority of *Projection* over *Linear* traversal (Fig. 4, Fig. 7) also demonstrates the need for localized approximations of latent space geometry for complex image features. This is in contrast to past traversal studies [4, 14, 16, 28, 31] that found global linear directions to suffice for simple scalar attributes.

A consideration in all image synthesis works is the balance between perceptual quality based on human judgment, and quantitative optimization and analysis. In the application of faces, the user may have his/her own internal tradeoff curve between identity preservation and image diversity. Our method offers a principled way to explore different points on this curve by tuning the β parameters (see Supplementary). Image perception also factors into the embedding functions used to measure image changes.

GAN latent spaces are not all alike, and each requires different considerations. Faces are easier to model than living rooms, because the latter are a composition of many discrete objects interacting with one another. As a result, we found the face latent space and traversals to be smoother. Our living room traversals often exhibit large perceptual "jumps" due to discontinuities in latent space (see Supplementary). The complexity of a distribution also affects the degree

of correlation between attributes. As Fig. 6 shows, it is not always possible to exactly fix a particular region of a living room while obtaining enough diversity elsewhere due to entangled features. Different regions of the latent space are also not alike. We found that high-likelihood regions produce the most realistic images and diverse traversals. Thus, the biases of the generative model have a direct effect on how well our method performs for a given image.

Our method takes a local view of the latent space to identify good traversal directions. However, as our results in Sec. 4.5 suggest, there are benefits to taking a global view. Global linear models are likely better for attributes that are discrete, such as 'wearing eyeglasses,' or approximately discrete for a large majority of samples like gender. For such attributes, local gradients in latent space can be near zero and swamped by noise. Another limitation of a local view is that gradients are undefined near sharp discontinuities in the latent space. We did not find this to be a decisive issue for faces, but did notice perceptual 'jumps' in the living room scenes during traversals (see Supplementary for traversal strips). However, we note that our framework could be extended to use both global and local directions per traversal step, which we leave for future work.

5.1 Ethics

Fairness: As in past work [4] we observed bias in StyleGAN's face distribution: Caucasian faces are most likely to be generated. This bias also affects trajectory quality, with light-skinned seed faces producing more diverse trajectories than dark-skinned ones. Biases in fixed and changing functions that use learning models also affect results. One example are face recognition models, like the one we used in our experiments to fix identity, which are known to have gender and ethnicity biases. To reduce bias one will want to train GANs and any learned models on rich and diverse datasets. **Fake portrayals**: GANs could be used to generate fake images of individuals under different conditions. This could include the case where the image of the face of a real person is projected onto the GAN latent space and then manipulated.

6 Conclusion

We presented a simple, principled and versatile method designed to explore a generative model's latent space to produce sets of synthetic samples where one group of multidimensional features is held constant while another is varied as much as possible. We demonstrated traversal results on several features that previous works are not capable of handling: landmark locations, pixels within regions, frequency information, and facial identity as measured by a deep neural network. Our experiments show the need for modeling local geometry of latent spaces for high-dimensional features. Understanding the complex nature and geometry of the latent space of image generators is a fascinating question which we have only started to explore.

References

1. Mediapipe. https://github.com/google/mediapipe
2. Antipov, G., Baccouche, M., Dugelay, J.L.: Face aging with conditional generative adversarial networks. In: 2017 IEEE international conference on image processing (ICIP). pp. 2089–2093. IEEE (2017)
3. Arvanitidis, G., Hansen, L.K., Hauberg, S.: Latent space oddity: on the curvature of deep generative models. arXiv preprint arXiv:1710.11379 (2017)
4. Balakrishnan, G., Xiong, Y., Xia, W., Perona, P.: Towards causal benchmarking of bias in face analysis algorithms. In: European Conference on Computer Vision. pp. 547–563. Springer (2020)
5. Balestriero, R., Paris, S., Baraniuk, R.: Max-affine spline insights into deep generative networks. arXiv preprint arXiv:2002.11912 (2020)
6. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Towards open-set identity preserving face synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6713–6722 (2018)
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019)
8. Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., Smagt, P.: Metrics for deep generative models. In: International Conference on Artificial Intelligence and Statistics. pp. 1540–1550. PMLR (2018)
9. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8789–8797 (2018)
10. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
11. Ghojogh, B., Karray, F., Crowley, M.: Eigenvalue and generalized eigenvalue problems: Tutorial. arXiv preprint arXiv:1903.11240 (2019)
12. Goetschalckx, L., Andonian, A., Oliva, A., Isola, P.: Ganalyze: Toward visual definitions of cognitive image properties. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5744–5753 (2019)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. vol. 27 (2014)
14. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: Advances in Neural Information Processing Systems. pp. 9841–9850 (2020)
15. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing **28**(11), 5464–5478 (2019)
16. Jahanian*, A., Chai*, L., Isola, P.: On the "steerability" of generative adversarial networks. In: International Conference on Learning Representations (2020)
17. Karras, T., Laine, S., Aila, T.: 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)
18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. arXiv preprint arXiv:1912.04958 (2019)

19. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
20. Kocaoglu, M., Snyder, C., Dimakis, A.G., Vishwanath, S.: Causalgan: Learning causal implicit generative models with adversarial training. In: International Conference on Learning Representations (2018)
21. Kuhnel, L., Fletcher, T., Joshi, S., Sommer, S.: Latent space non-linear statistics. arXiv preprint arXiv:1805.07632 (2018)
22. Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., Ranzato, M.: Fader networks: Manipulating images by sliding attributes. In: 31st Conference on Neural Information Processing Systems (NIPS 2017). pp. 5969–5978 (2017)
23. Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., Wen, S.: Stgan: A unified selective transfer network for arbitrary image attribute editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3673–3682 (2019)
24. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
25. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: International conference on machine learning. pp. 2642–2651. PMLR (2017)
26. Or-El, R., Sengupta, S., Fried, O., Shechtman, E., Kemelmacher-Shlizerman, I.: Lifespan age transformation synthesis. In: European Conference on Computer Vision. pp. 739–755. Springer (2020)
27. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2337–2346 (2019)
28. Plumerault, A., Borgne, H.L., Hudelot, C.: Controlling generative models with continuous factors of variations. In: International Conference on Learning Representations (2020)
29. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
30. Shao, H., Kumar, A., Thomas Fletcher, P.: The riemannian geometry of deep generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 315–323 (2018)
31. Shen, Y., Gu, J., Tang, X., Zhou, B.: 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)
32. Shen, Y., Luo, P., Yan, J., Wang, X., Tang, X.: Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 821–830 (2018)
33. Shen, Y., Zhou, B., Luo, P., Tang, X.: Facefeat-gan: a two-stage approach for identity-preserving face synthesis. arXiv preprint arXiv:1812.01288 (2018)
34. Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G.: Gan-control: Explicitly controllable gans. arXiv preprint arXiv:2101.02477 (2021)
35. Tran, L., Yin, X., Liu, X.: Disentangled representation learning gan for pose-invariant face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1415–1424 (2017)
36. Tzelepis, C., Tzimiropoulos, G., Patras, I.: Warpedganspace: Finding non-linear rbf paths in gan latent space. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6393–6402 (2021)

37. Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep feature interpolation for image content changes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7064–7073 (2017)
38. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the gan latent space. In: International Conference on Machine Learning. pp. 9786–9796. PMLR (2020)
39. Wang, B., Ponce, C.R.: A geometric analysis of deep generative image models and its applications. In: International Conference on Learning Representations (2021)
40. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)
41. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)
42. Xiao, T., Hong, J., Ma, J.: Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: Proceedings of the European conference on computer vision (ECCV). pp. 168–184 (2018)
43. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. International Journal of Computer Vision pp. 1–16 (2021)
44. Yang, H., Chai, L., Wen, Q., Zhao, S., Sun, Z., He, S.: Discovering interpretable latent space directions of gans beyond binary attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12177–12185 (2021)
45. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3990–3999 (2017)
46. Zhu, J., Feng, R., Shen, Y., Zhao, D., Zha, Z.J., Zhou, J., Chen, Q.: Low-rank subspaces in gans. Advances in Neural Information Processing Systems **34** (2021)