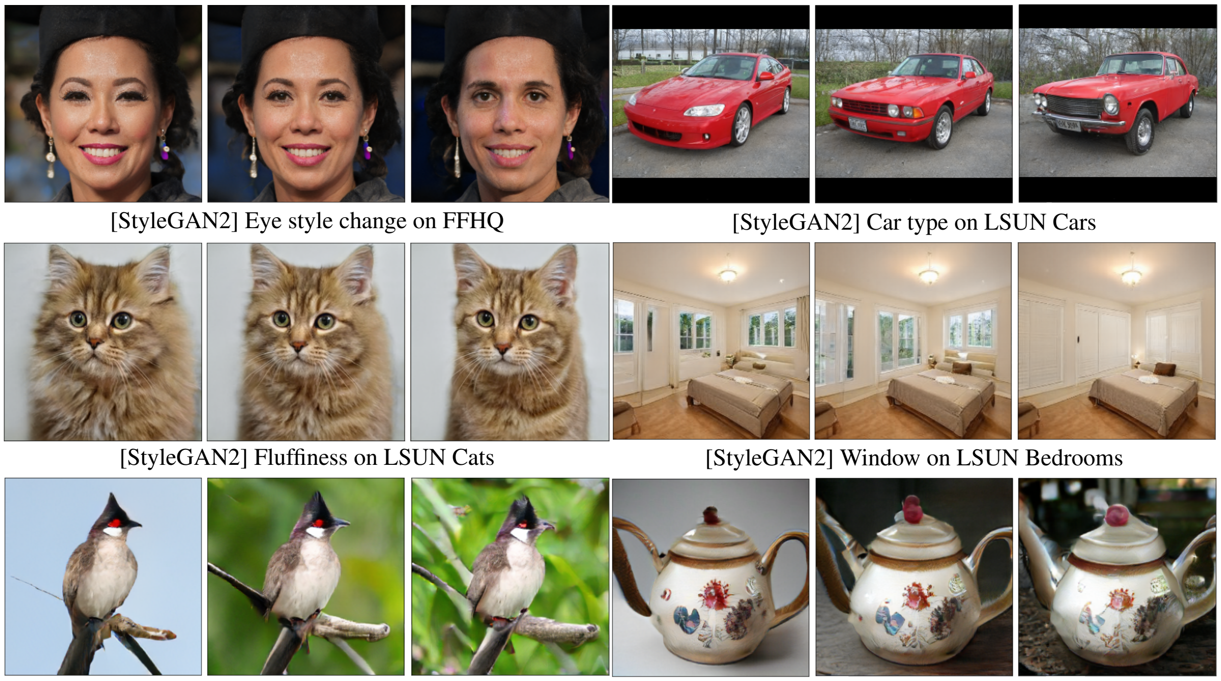
LatentCLR: A Contrastive Learning Approach for Unsupervised Discovery of Interpretable Directions

Oğuz Kaan Yüksel[1,†] Enis Simsar[2,3,†] Ezgi Gülperi Er[3] Pinar Yanardag[3]
[1]EPFL [2]TUM [3]Bogazici University
oguz.yuksel@epfl.ch enis.simsar@tum.de ezgi.er@boun.edu.tr yanardag.pinar@gmail.com



[StyleGAN2] Eye style change on FFHQ

[StyleGAN2] Car type on LSUN Cars

[StyleGAN2] Fluffiness on LSUN Cats

[StyleGAN2] Window on LSUN Bedrooms

[BigGAN] Background removal on ImageNet *Bulbul* class

where the same direction is transferred to the *Teapot* class.

Figure 1: Interpretable directions discovered in StyleGAN2 [10] and BigGAN [1] (left and right images are obtained by moving the latent code of the image in the middle towards negative or positive direction found by LatentCLR). Our directions are transferable, such as background removal learned on *Bulbul* class also removes the background on *Teapot* class.

Abstract

Recent research has shown great potential for finding interpretable directions in the latent spaces of pre-trained Generative Adversarial Networks (GANs). These directions provide controllable generation and support a wide range of semantic editing operations such as zoom or rotation. The discovery of such directions is often performed in a supervised or semi-supervised fashion and requires manual annotations, limiting their applications in practice. In comparison, unsupervised discovery enables finding subtle directions a priori hard to recognize. In this work, we propose a contrastive-learning-based approach for discovering semantic directions in the latent space of pretrained GANs in a self-supervised manner. Our approach finds semantically meaningful dimensions compatible with state-of-the-art methods.

1. Introduction

Generative Adversarial Networks (GANs) [5] are powerful image synthesis models that revolutionized generative modeling in computer vision. They have been widely used for various visual tasks due to their success at synthesizing high-quality images. Image generation [31], image manipulation [28], image denoising [27, 12], image super-resolution [24] and domain translation [32] are included in

[†] Equal contribution. Author ordering determined by a coin flip.

some of the many creative uses of generative models.

Up until recently, GAN models are typically interpreted as black-box models without the ability to control the generated images. A certain amount of control can be obtained by training conditional models [13] and generating images with specified attributes. However, this approach requires labeled data and only offers a limited control that depends on the available supervised information. Another line of work aims to design models to produce a more disentangled latent space [3] where each latent dimension controls a particular attribute. However, what knowledge GANs learn in the latent representation and how to use these representations to manipulate images is still an ongoing research question. The first attempts to explicitly control the underlying generation process of GANs include simple approaches such as modifying the latent code of the images [17] or by interpolating latent vectors [9]. Recently, several approaches are proposed to explore the structure of latent space in GANs in a more principled way [21, 7, 26, 8]. The majority of these works discover domain-agnostic interpretable directions such as *zoom-in, rotation* or *translation* [16, 8] while others proposed frameworks to find domain-specific directions such as changing *gender, age*, or *expression* on face images [21]. Usually, the obtained directions are used for modifying image semantics by moving the latent code towards the identified direction by a certain amount in order to increase or decrease the property of interest.

In this paper, we introduce LatentCLR, a learning-based approach using a self-supervised contrastive objective to find interpretable directions in GANs. In particular, we use the *differences* caused by an edit operation on the feature activations to optimize identifiability of each direction in order to find a diverse set of semantically meaningful manipulations. Our contributions are as follows:

- We propose a contrastive-learning based on feature divergences to discover interpretable directions in the latent space of pre-trained GAN models such as StyleGAN2 [10] and BigGAN [1]. To the best of our knowledge, we are the first to introduce the discovery of interpretable directions using contrastive learning.
- We demonstrate that our method is able to find several diverse and fine-grained directions on a variety of datasets and we demonstrate that the obtained directions are highly transferable to other ImageNet [18] classes.
- We publicly share our implementation to encourage further research in this area: https://github.com/catlab-team/latentclr.

The rest of this paper is organized as follows. Section 2 discusses related work in latent space manipulation. Section 3 introduces our contrastive framework. Section 4 presents our quantitative and qualitative results. Section 5 discusses the limitations and broader impact of our work and Section 6 concludes the paper.

2. Related Work

In this section, we address the related work on generative adversarial networks, as well as unsupervised and supervised latent space manipulation methods.

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are two-part networks that use deep learning methods to model the real-world to the generative space [5] where the main goal is to model the image space in such a way that it generates images that are indistinguishable from those in the dataset. The adversarial part of the network tries to detect whether the produced image is from the training dataset or a generated one, while the generative part tries to create images that are similar to the dataset. Internal mapping allows generative networks to create new realistic images from random noise vectors known as latent vectors.

StyleGAN [9] and StyleGAN2 [10] are among popular GAN approaches that are capable of generating high-resolution images. They use a mapping network consists of an 8-layer multilayer perceptron which aims to fit the input latent code onto an intermediate latent space. Another popular GAN model is BigGAN [1], a large-scale model trained on ImageNet [19]. Similar to StyleGAN, it also utilizes the intermediate layers by taking the latent vector as input, also called *skip-z* inputs. It also uses a class vector as input and its conditional architecture can generate images in a variety of categories from ImageNet.

In this paper, we work on pre-trained StyleGAN2 and BigGAN models to discover semantically meaningful directions in latent space.

2.2. Latent Space Manipulation

Recently, several strategies for manipulating the latent structure of pretrained GANs have been proposed. These methods manipulate images in a variety of ways by editing the latent code and can be classified into two groups.

Supervised Latent Space Manipulation Supervised approaches typically use pre-trained classifiers to guide an optimization based learning to discover interpretable directions that specifically manipulate directions of interest. InterfaceGAN [21] is a supervised approach that benefits from labeled data including *gender, facial expression* and *age*. It trains a binary Support Vector Machine (SVM) [14] on labeled data and interprets the normal vector of the obtained hyper-plane as a latent direction. [4] finds directions for cognitive image properties in a pretrained BigGAN model using an *assessor* function which is an externally trained classifier. The feedback from the assessor guides their optimization process and the resulting optimal direction enables manipulation of desired cognitive attributes.

Unsupervised Latent Space Manipulation [26] suggests an unsupervised method for discovering meaningful directions based on a classifier-based approach. Given a particular manipulation, the classifier tries to detect which particular direction is applied. At the end of the optimization process, their method ends up learning disentangled directions. Thus, their method results in that aims to increase distinguishability. Their method aims to define a set of directions that corresponds to various image transformations. Ganspace [7] is a sampling-based unsupervised method that randomly samples latent vectors from the intermediate layers of BigGAN and StyleGAN models. Then they propose to use Principal Component Analysis (PCA)[29] to find principal components which are interpreted as semantically meaningful directions. The principal components result in a variety of useful manipulations including *zoom, rotation* in BigGAN, or changing *gender, hair color, or age* on StyleGAN models. SeFa [22] follows a related approach using a closed-form solution that specifically optimizes the intermediate weight matrix of the pre-trained GAN model. They obtain interpretable directions in the latent space by computing the eigenvectors on the first projection step matrix and selecting the eigenvectors associated with the largest eigenvalues. [8] suggests an approach to take advantage of task-specific edit functions. They begin by applying an editing operation to the original image, such as zooming, and minimize the distance between the original image and the edited image in order to learn a direction that results in the desired editing operation. However, their approach is limited to the available set of editing functions.

We believe that the closest methods to our work are Ganspace [7] and SeFa [22] methods which we extensively compare in Section 4.

3. Methodology

In this section, we first introduce preliminary information contrastive learning and then introduce our framework.

3.1. Contrastive Learning

Contrastive learning has recently become popular due to leading state-of-the-art results in several unsupervised representation learning tasks. It aims to learn representations by contrasting positive pairs against negative pairs [6] and used in several computer vision tasks including data augmentation [2], random cropping and flipping [15] or diverse scene generation [25]. The key idea of contrastive learning is to push the representations of similar pairs to be close while dissimilar pairs to be far apart.

In this paper, we adopt the SimCLR [2] framework for contrastive learning. SimCLR consists of four major components: a stochastic data augmentation method producing positive pairs $(\mathbf{x}, \mathbf{x}^+)$, an encoding network f that extracts representation vectors out of augmented samples, a small projector head g that maps representations to the loss space, a contrastive loss function ℓ that enforces separation between positive and negative pairs. Given a random minibatch of N samples, SimCLR produces N positive pairs by utilizing the given data augmentation method. For all positive pairs, the remaining $2(N-1)$ augmented samples are treated as negative examples. Let $\mathbf{h}_i = f(\mathbf{x}_i)$ be the representations of all $2N$ samples and $\mathbf{z}_i = g(\mathbf{h}_i)$ be the projections of these representations. Then, SimCLR considers the average of the *NT-Xent* loss [23, 15] over all positive pairs $(\mathbf{x}_i, \mathbf{x}_j)$:

$$\ell(\mathbf{x}_i, \mathbf{x}_j) = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \quad (1)$$

where $\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$ is the cosine similarity function, $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if and only if $k = i$, and τ is the temperature parameter. Both networks f and g are trained together.

Intuitively, g learns a mapping to a space where cosine similarity represents the semantic similarity and NT-Xent objective encourages *identifiability* of positive pairs among all other negative examples. This, in turn, forces f to learn representations that are invariant to the given data augmentations, up to a nonlinear mapping and cosine similarity.

3.2. Latent Contrastive Learning (LatentCLR)

In this paper, we aim to use a contrastive learning approach to find semantically meaningful directions in latent space. Similar to [7, 22, 26], we limit ourselves to the unsupervised setting, where we aim to identify such edit directions without any external supervision utilized in works such as [4, 21, 8]. Given a target representation space, our intuition is to work on the *differences* caused by an edit operation rather than the feature activations and optimize an *identifiability-based* heuristic to find a diverse set of interpretable directions, similar to [26]. That is, we enforce each edit operation to yield directions that are easily separated and identified. However, our formulation does not utilize an additional classifier but, instead, a self-supervised strategy using contrastive learning as described below.

For a given batch of random latent codes, consider simultaneous distinct edit operations and corresponding feature divergences. We label divergence pairs that resulted from the same edit operation *positive*, to encourage consistency, and label others *negative*, to encourage disentanglement. We optimize edit operations to minimize a contrastive loss, detailed in this section defined over these positive and negative pairs.

Assume a pretrained GAN, expressed as a mapping function $\mathbf{G} : \mathcal{Z} \to \mathcal{X}$ where $\mathcal{Z}$ is the latent space, usually associated with a prior distribution such as multivariate Gaussian, and $\mathcal{X}$ is the target image domain. Given a latent code $\mathbf{z}$ and its generated image $\mathbf{x} = \mathcal{G}(\mathbf{z})$, we seek to find edit directions $\Delta\mathbf{z}$ such that the image $\mathbf{x}' = \mathcal{G}(\mathbf{z} + \Delta\mathbf{z})$ has semantically meaningful changes over $\mathbf{x}$ while still preserving the identity of $\mathbf{x}$.

Objective function: For each latent code $\mathbf{z}_i$ in the mini-batch of size N, we compute K distinct edited latent codes $\mathbf{z}_i^k$ and the associated intermediate feature representations $\mathbf{h}_i^k = \mathcal{G}_f(\mathbf{z}_i^k)$, where $\mathcal{G}_f$ applies feed-forward until a target layer f. Next, we calculate the feature divergences $\mathbf{f}_i^k = \mathbf{h}_i^k - \mathcal{G}_f(\mathbf{z})$ and consider the following loss defined for each edited latent code $\mathbf{z}_i^k$:

$$\ell(\mathbf{z}_i^k) = -\log \frac{\sum_{j=1}^{N} \mathbb{1}_{[j \neq i]} \exp\left(\text{sim}(\mathbf{f}_i^k, \mathbf{f}_j^k)/\tau\right)}{\sum_{j=1}^{N} \sum_{l=1}^{K} \mathbb{1}_{[l \neq k]} \exp\left(\text{sim}(\mathbf{f}_i^k, \mathbf{f}_j^l)/\tau\right)}$$

The intuition behind our objective function is as follows. All the feature divergences obtained with the same latent edit operation as $\mathbf{z}_i^k$, viewed as *positive* pairs and contribute to the numerator. All other divergences obtained with a different edit operation $l \neq k$ is considered as *negative* pairs and contributes to the denominator. This can be seen as a generalization of NT-Xent loss (Eq. 1), where we have N-tuples of positive labeled group of augmented samples. With this generalized contrastive loss, we enforce each latent edit operation to have a consistent but also orthogonal effect on the features.

Direction models: Our approach contains a learnable component called *direction model(s)*. The direction model is a mapping $\mathcal{D} : \mathcal{Z} \times \mathbb{R} \to \mathcal{Z}$, that takes latent codes together with a desired edit *magnitude* and outputs edited latent codes, i.e., $\mathcal{D} : (\mathbf{z}, \alpha) \to \mathbf{z} + \Delta\mathbf{z}$, where $\|\Delta\mathbf{z}\| \propto \alpha$.

Choice of direction model We consider three alternative methods for the choice of direction model; *global, linear* and *non-linear*, defined as follows:

- *Global.* We learn a fixed direction θ irrespective of the latent code $\mathbf{z}$.

$$\mathcal{D}(\mathbf{z}, \alpha) = \mathbf{z} + \alpha \frac{\theta}{\|\theta\|}$$

- *Linear.* We learn a matrix $\mathbf{M}$ to output a conditional direction on the latent code z, representing a linear dependency.

$$\mathcal{D}(\mathbf{z}, \alpha) = \mathbf{z} + \alpha \frac{\mathbf{Mz}}{\|\mathbf{Mz}\|}$$

- *Nonlinear.* We learn a multi-layer perceptron, represented with $\mathbf{NN}$, to represent an arbitrarily complex dependency between direction and the latent code.

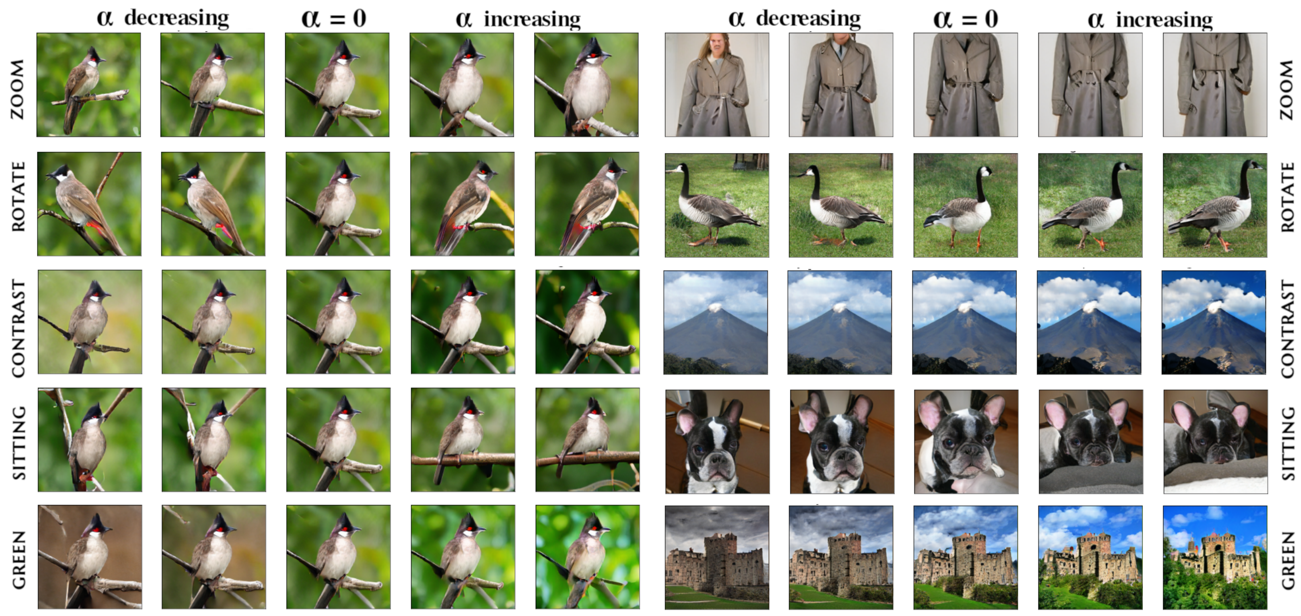
$$\mathcal{D}(\mathbf{z}, \alpha) = \mathbf{z} + \alpha \frac{\mathbf{NN}(\mathbf{z})}{\|\mathbf{NN}(\mathbf{z})\|}$$

For all options, we consider ℓ_2 normalization and, therefore, magnitudes we specify with α correspond to direct ℓ_2 distances from the latent code.

The first option, *Global*, in principle the most limited, since it can only find fixed directions and thus edits images without considering the latent code z. However, we observe that it is still able to capture common directions such as *zoom, rotation* or *background removal* on BigGAN [1]. The second option, *Linear* is able to generate *conditional* directions given a latent code z, however it is still limited for capturing more fine-grained directions. The third option is an extension of *Linear* direction model where we use a neural network that models the dependency between the direction and the given latent code. In our experiments, we observe that the neural network based *Nonlinear* direction model captures a wide range of interpretable directions as well as discovering class-specific directions such as adding or removing *tongue* on *Husky* class or *adding flowers to branches* in *Bulbul* class [18] (see Figure 3). For the rest of this paper, we use *Nonlinear* direction model in our experiments.

Choice of K: For learning K number of different directions, we use K copies of the same direction model. We observe that using too many directions leads to repetitive directions, a similar observation that is also made by [26]. For BigGAN, we used $K = 32$ directions since its latent space is 128-dimensional and most of the interpretable directions such as *zoom, rotation* or *translation* can be obtained with a relatively small number of directions. For StyleGAN2 [10], we used $K = 100$ directions since the latent space is 512-dimensional and there are various diverse directions especially on complex datasets such as FFHQ [9].

Utilizing Layer-wise Styles: The layer-wise structure of StyleGAN2 and BigGAN models can be used for fine-grained editing, as pointed out by [7] where different semantics are controlled with different layer groups. By limiting the application of PCA-identified directions to groups of layers instead of all layers, they achieve more restricted semantic changes. However, how their initial formulation



(a) (b)

Figure 2: (a) Directions for general image editing operations such as *zoom*, or *rotation* discovered from ImageNet Bulbul class where we shift the latent code towards a particular direction with increasing or decreasing α. (b) Transferred directions from *Bulbul* class to various other ImageNet classes.

considers the effect of intermediate layers is not clear, as PCA-identified directions are usually found in previous layers ($\mathcal{W}$-space for StyleGAN2 and outputs of the first linear layer of BigGAN512-deep), before any intermediate layer. Note that the authors report that using later intermediate layers yields lesser control than those given by the aforementioned initial layers. Similarly, [22] can identify directions associated with each layer group and apply directions to a restricted set of layers.

In this study, we also observe that fine-grained semantics can be controlled with different layer blocks. As explained in Section 3, contrary to [7], our learning method can optionally consider only the effects of a selected subset of layers while finding directions. Combined with the improvements over fixed directions, our method can identify fine-grained semantic directions. And contrary to [22], we can find orthogonal directions and fuse overlapping effects of distinct layers in a single feature activation space.

4. Experiments

We evaluate the proposed method to discover semantically meaningful directions on several models and datasets. We apply the proposed model on BigGAN [1] and StyleGAN2 [10] models on a wide range of datasets including human faces (FFHQ) [9], LSUN Cats, Cars, Bedrooms, Church and Horse datasets, [30] and ImageNet [18]. We also compare our method to state-of-the-art unsupervised methods [7, 22]*, and run several qualitative and quantitative experiments to demonstrate the effectiveness of our approach. Next, we present our BigGAN and StyleGAN2 results and then discuss our experimental setup.

4.1. Results on BigGAN

We evaluate our approach on pre-trained BigGAN [1] model which is conditionally trained on 1000 ImageNet [18] classes. We trained our model on an arbitrary class, *Bulbul*, and obtained $K = 32$ directions. We investigated the following questions:

Q1: Can we discover semantically meaningful directions for general image editing operations such as *zoom*, or *rotation*? Our visual analysis shows that our model is able to recover several semantically meaningful directions such as *zoom, rotate, contrast* as well as some finer-grained features such as *background removal, sitting* or *green background* (see Figure 1 and Figure 2 (a)). As can be seen from the visuals, our method is able to manipulate the original image (denoted with α=0) by moving the latent code toward (increasing α) and backward (decreasing α) the interpretable direction.

*Note that we had to omit another competitor [26] since their BigGAN and StyleGAN2 models are using different pretrained models and producing the same baseline images for comparison is not possible.

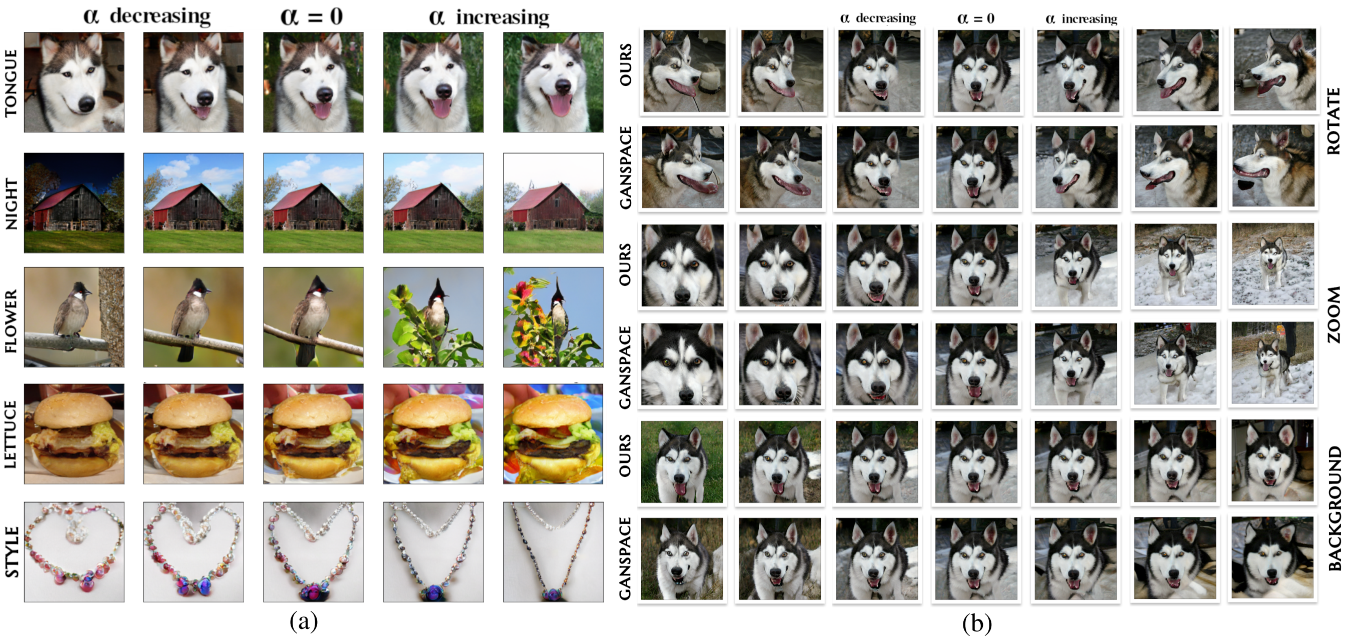


Figure 3: (a) Class-specific directions discovered by our method in several ImageNet classes on BigGAN model. (b) A comparison of *rotate, zoom* and *background change* directions between our method and Ganspace [7].

Q2: Are the editing directions we obtained *transferable* to other classes? After verifying that the directions obtained for *Bulbul* class are able to manipulate the latent codes for several semantic directions, we investigated how transferrable are the discovered directions on other ImageNet classes. Our visual analysis shows that the directions learned from *Bulbul* class is applicable to a diverse range of ImageNet classes, and able to zoom a *Trenchcoat*, rotate a *Goose*, apply contrast to *Volcano*, add greenness to *Castle* classes (see Figure 2 (b)) as well as removing the background from a *Teapot* object (see Figure 1, bottom right image). An interesting transferred direction is *Sitting* (see Figure 2 (a)) which manipulates the latent code so that the bird in Bulbul class sits on a tree branch with increasing α. We observed that when this direction is applied to *Bulldog* class, it is also able to make the dog stand up or sit down based on how we move in positive or negative directions (see Figure 2 (b)).

Q3: Can we discover semantically meaningful but *fine-grained* editing directions tailored towards particular classes? Next, we investigate whether we can discover unique directions when we train our model on different ImageNet classes. Figure 3 (a) shows some directions discovered by our model, such as removing or adding *tongue* on *Husky* class, changing the *time of the day* in *Barn* class, adding *flowers* in *Bulbul* class, adding or removing *lettuce* on *Cheeseburger* class, or changing the *style* of the necklace in *Necklace* class.

Q4: How successful are the obtained directions comparing to other methods? We visually compare[†] the directions obtained by our method with Ganspace[7] using *Husky* class. For each direction in Ganspace, we use the provided parameters as given in their open-source implementation[‡]. For both methods, we use the same seed image (shown with $\alpha = 0$) and move the latent code towards the direction (with increasing α) and far away from the direction (with decreasing α). We used the original α settings (i.e. the *sigma* setting) for each direction as provided in the implementation of Ganspace in order to avoid any bias that can be caused by tuning the parameter.

We observe that both methods achieve similar manipulations for *zoom, rotation*, and *background change* directions while our method causes less entanglement. For instance, we notice that Ganspace tends to increase *tongue* with *rotation* or add background objects with increasing *zoom*.

Q5: How *semantically meaningful* are the obtained $K = 32$ directions?

In order to understand how the directions found by our method are aligned with human perception, we conduct a

[†] Note that we were not able to compare with the SeFa method since they do not provide BigGAN directions in their open-source model: https://github.com/genforce/sefa.

[‡] https://github.com/harskish/ganspace

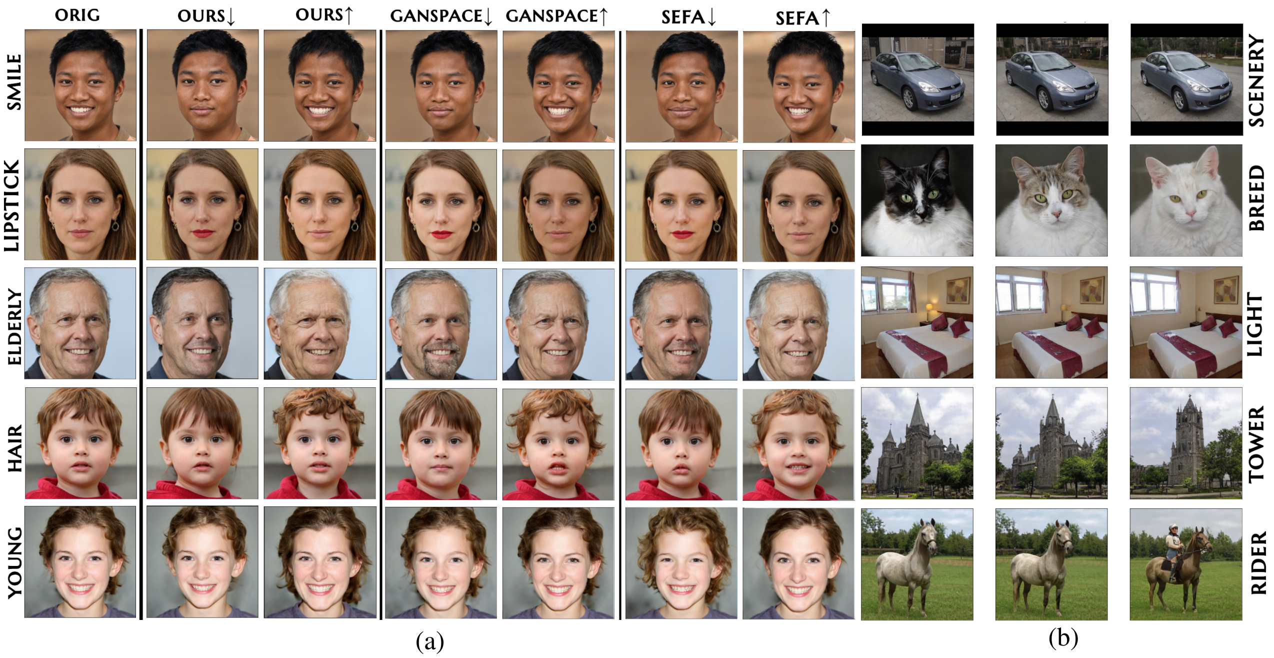


Figure 4: (a) Comparison of manipulation results on FFHQ dataset with Ganspace [7] and SeFa [22] methods. The leftmost image represents the original image, while images denoted with ↑ and ↓ represent the edited image moved in the positive or negative direction. (b) Directions discovered by our method in LSUN [30] datasets.

user study on Amazon's Mechanical Turk platform where each participant is shown a randomly selected 10 images out of a set of 100 randomly generated image (where the original image is displayed in the middle, and $-\alpha$ and $+\alpha$ values are displayed on the left and right side, respectively). Following the same approach of [22] we ask $n = 100$ users the following questions: *”Question 1: Do you think there is an obvious content change on the left and right images comparing to the one in the middle?”* and *”Question 2: Do you think the change on the left and right images comparing to the one in the middle is semantically meaningful?”*. Each question is associated with *Yes/Maybe/No* options and the order of the questions is also randomized. Our user study shows that for *Question 1*, participants answered *Yes* to an average of 17.43 directions, *Maybe* to an average of 7.43 directions and *No* to an average 5.75 directions. Similarly, for *Question 2* we obtained *Yes* to an average of 14.37 directions, *Maybe* to an average of 10.03 directions and *No* to an average 6.21 directions. These results indicate that out of $K = 32$ directions, participants found 82% semantically meaningful, and 80.59% directions containing obvious content change.

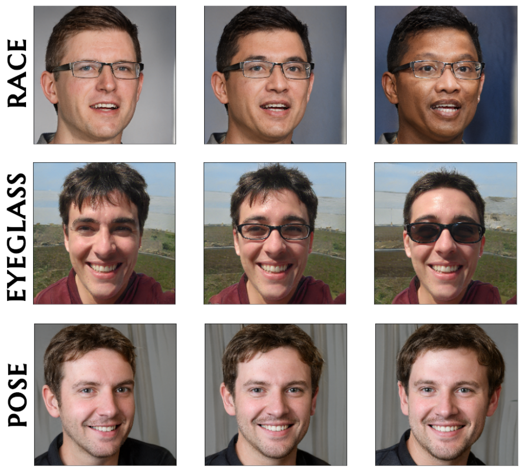


Figure 5: Directions discovered by our method in FFHQ dataset with StyleGAN2.

4.2. Results on StyleGAN2

We apply our method on StyleGAN2 on a wide range of datasets and compare our results against state-of-the-art unsupervised[§] methods Ganspace [7] and SeFa [22].

We first visually examined the directions found by our method on various datasets including FFHQ [9], and LSUN Cars, Cats, Bedroom, Church and Horse datasets [30] (see Figure 1 and Figure 4 (b)). Our method is able to dis-

[§] We weren't able to compare our results with supervised method InterfaceGAN since it currently does not have StyleGAN2 support, see http://github.com/genforce/interfacegan/issues/52

Model	Ganspace	SeFa	Ours
↑ Smile	0.99 ±0.11	0.89 ±0.31	0.99 ±0.11
↑ Age	0.32 ±0.12	0.38 ±0.15	0.42 ±0.13
↑ Lipstick	0.58 ±0.49	0.55 ±0.49	0.66 ±0.47
↓ Smile	0.05 ±0.21	0.50 ±0.50	0.11 ±0.32
↓ Age	0.23 ±0.08	0.23 ±0.06	0.23 ±0.07
↓ Lipstick	0.35 ±0.47	0.35 ±0.48	0.36 ±0.48

Table 1: Rescoring Analysis on three attributes in FFHQ dataset on StyleGAN2.

cover several fine-grained directions such as changing *car type* or *scenery* in LSUN Cars, changing *breed* or adding *fur* in LSUN Cats, adding *windows* or turning on *lights* in LSUN Bedrooms, adding *tower* details to churches in LSUN Church and adding *riders* on LSUN Horse datasets.

Next, we compare how the directions found on FFHQ differ across methods. Figure 4 (a) shows the visual comparison between several directions that are commonly found by all methods, including *Smile, Lipstick, Elderly, Curly Hair* and *Young* directions. As can be seen from the visuals, all methods perform similarly and able to manipulate the images towards the desired attributes. Figure 5 illustrates specific directions *Race, Eyeglass* and *Pose*.

In order to understand how our method compares against the competitors in a quantitative fashion, we performed a re-scoring analysis following [22] where we use predictors to understand whether the manipulations properly altered images towards desired attributes[¶].

We used attribute predictors released by StyleGAN2 [10] for *Smile* and *Lipstick* directions as these are the only two attributes with an available predictor and at the same time commonly found by all methods[‖]. We used an off-the-shelf age predictor [20] for *Age* direction. Table 1 shows our results for re-scoring analysis for three attributes: *Age*, *Lipstick* and *Smile* for negative (denoted with ↓) and positive (denoted with ↑) directions. In order to see how the scores change, we first randomly generate 500 images for each property. The average score obtained by the predictors for *Lipstick* property is 0.43 ±0.5, average *Age* score is 0.27 ±0.1 and average *Smile* score is 0.74 ±0.43. When we move the latent codes towards the negative direction, we observe that our method as well as Ganspace and SeFa methods decrease the scores within similar ranges in Age and Lipstick properties. We observe that when moving towards negative direction for *smile* property, both our method and Ganspace are able to reduce the score significantly (from 0.74 to 0.11 and 0.05, respectively) while SeFa reduces the score from an average of 0.74 to 0.50. On the other hand, when we move the latent codes towards the positive direction all methods obtain comparable results.

4.3. Experimental Setup

For BigGAN experiments, we pick *batch size* = 16, K = 32 directions, *output resolution* = 512, *truncation* = 0.4, *feature layer* = *generator.layers.4* and train the models for 3 epochs (each epoch is 100k iterations) which takes approximately 20 minutes. For StyleGAN2 experiments, we pick *batch size* = 8, K = 100 directions, *truncation* = 0.7, *feature layer* = *conv1* and train StyleGAN2 models 5 epochs (each epoch is 10k iterations) which takes approximately 12 minutes. We use PyTorch framework for our experiments and used two NVIDIA Titan RTX GPUs.

In order to avoid any biases between competitor methods Ganspace and SeFa, we used the same set of StyleGAN2 layers obtained from Ganspace [7]. We also point out that slight differences in the re-scoring analysis or visuals might be caused due to applying different magnitudes of change across methods since they are not directly convertible across methods. In order to minimize this effect, we use the same sigma values in Ganspace as specified for each direction in their public repository, and used $\{-3, +3\}$ for SeFa as provided in the official implementation.

5. Limitations

Our method uses a pre-trained GAN model as input, therefore it is only limited to manipulating GAN-generated images. However, it can be extended to real images by using GAN inversion methods [33] by encoding the real images into the latent space. Similar to any image synthesis tool, our method also shares similar concerns and dangers about misuse where it can be applied to images of people or faces for malicious purposes as discussed in [11].

6. Conclusion

In this study, we propose a framework that learns unsupervised directions using contrastive learning. Instead of discovering fixed directions such as [26, 7, 22], our method can discover non-linear directions in pre-trained StyleGAN2 and BigGAN models and leads to several diverse and semantically meaningful directions. We demonstrate the effectiveness of our approach on a wide range of models and datasets and compare it against state-of-the-art unsupervised methods. We publicly share our implementation at `https://github.com/catlab-team/latentclr`.

[¶] Since the attribute predictors used in [22] and [21] are not publicly available, we weren't able to compare for the same attributes. See `https://github.com/genforce/interfacegan/issues/37`.

[‖] We omit other common attributes *age* and *curly* hair since the predictors of StyleGAN2 since they output an average value of 0.99 for all methods, therefore a comparison was not possible.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

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

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[6] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

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

[8] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[10] 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.

[11] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[12] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K. Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis, 2019.

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

[14] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[16] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *arXiv preprint arXiv:2001.10238*, 2020.

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

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[20] Sefik Ilkin Serengil and Alper Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5. IEEE, 2020.

[21] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[22] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[23] Kihyuk Sohn. Improved deep metric learning with multiclass n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[24] Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020.

[25] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[26] 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.

[27] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson Lau. Spatial attentive single-image deraining with a high quality real rain dataset, 2019.

[28] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans, 2017.

[29] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

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

[31] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative ad-

versarial networks. *CoRR*, abs/1710.10916, 2017.

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[33] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020.