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]Technical University of Munich   [3]Boğaziçi University

oguz.yuksel@epfl.ch   enis.simsar@tum.de   ezgi.er@boun.edu.tr   yanardag.pinar@gmail.com
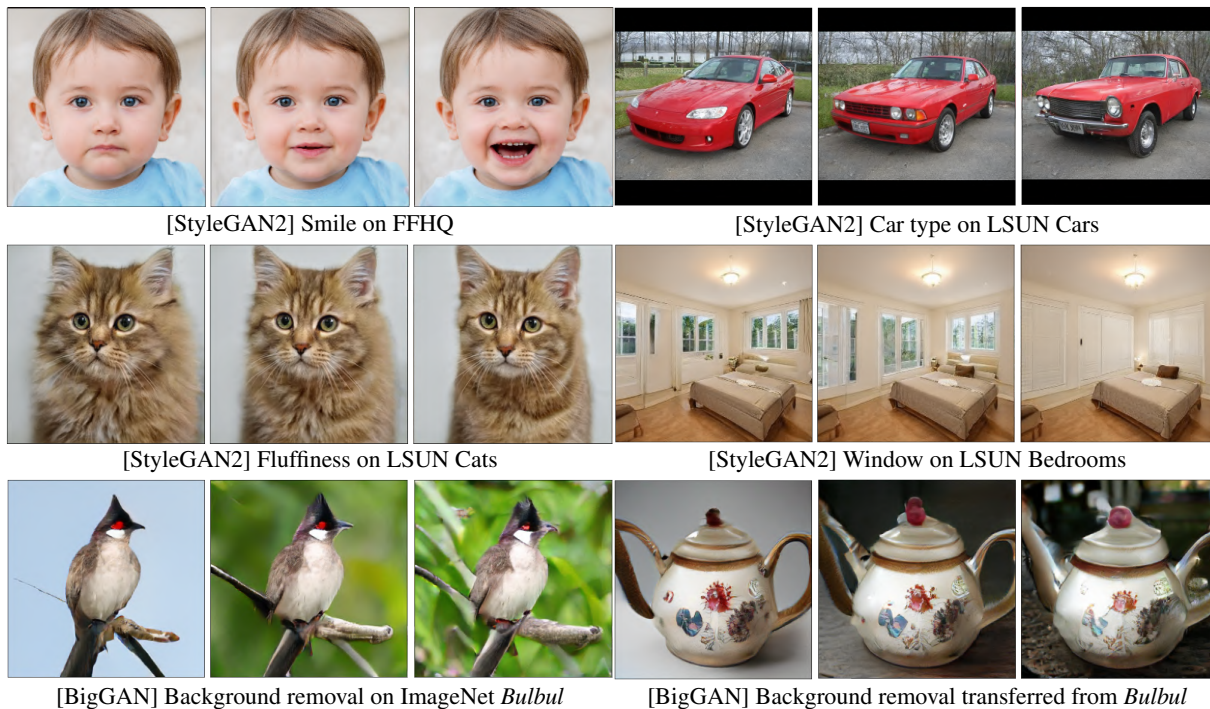
Figure 1: Interpretable directions discovered in StyleGAN2 [12] and BigGAN [2]. Left and right images of each triplet are obtained by moving the latent code of the image, shown in the middle, towards negative and positive directions, respectively. Our directions are transferable, such as background removal learned on *Bulbul* class.

Abstract

Recent research has shown that it is possible to find interpretable directions in the latent spaces of pre-trained Generative Adversarial Networks (GANs). These directions enable controllable image generation and support a wide range of semantic editing operations, such as zoom or rotation. The discovery of such directions is often done in a supervised or semi-supervised manner and requires manual annotations which limits their use in practice. In comparison, unsupervised discovery allows finding subtle directions that are difficult to detect a priori. In this work, we propose a contrastive learning-based approach to discover semantic directions in the latent space of pre-trained 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) [7] are powerful image synthesis models that have revolutionized generative modeling in computer vision. Due to their success in synthesizing high-quality images, they are widely used for various visual tasks, including image generation [38], image manipulation [35], de-noising [34, 15], upscaling image resolution [30], and domain translation [39].

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

Until recently, GAN models have generally been interpreted as black-box models, without the ability to control the generation of images. Some degree of control can be achieved by training conditional models such as [17] and changing conditions in generation-time. Another approach is to design models that generate a more disentangled latent space such as in InfoGAN [4] where each latent dimension controls a particular attribute. However, these approaches require labels and provide only limited control, depending on the granularity of available supervised information.

Albeit some progress has been done, the question of what knowledge GANs learn in the latent representation and how these representations can be used to manipulate images is still an ongoing research question. Early attempts to explicitly control the underlying generation process of GANs include simple approaches, such as modifying the latent code of images [22] or interpolating latent vectors [11]. Recently, several approaches have been proposed to explore the structure of latent space in GANs in a more principled way [10, 26, 9, 33, 21]. Most of these works discover domain-agnostic interpretable directions such as *zoom, rotation*, or *translation*, while other find domain-specific directions such as changing *gender, age* or *expression* on facial images. Typically, such methods either identify or optimize for directions and then shift the latent code in these directions to increase or decrease target semantics in the image.

In this paper, we introduce **LatentCLR**, an optimization-based approach that uses 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 the identifiability of each direction. Our contributions are as follows:

- We propose to use contrastive learning on feature divergences to discover interpretable directions in the latent space of pre-trained GAN models such as StyleGAN2 and BigGAN.

- We show that our method can find distinct and fine-grained directions on a variety of datasets, and that the obtained directions are highly transferable between ImageNet [24] classes.

- We make our implementation publicly available 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. Section 3 introduces our contrastive framework. Section 4 presents our quantitative and qualitative results. Section 5 discusses the limitations of our work and Section 6 concludes the paper.

2. Related Work

In this section, we introduce generative adversarial networks and discuss latent space manipulation methods.

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) consist of a generator and a discriminator for mapping the real world to the generative space [7]. The discriminator part of the network tries to detect whether images are from the training dataset or synthetic, while the generative part tries to generate images that are similar to the dataset. StyleGAN [11] and StyleGAN2 are among the popular GAN approaches that are capable of generating high-quality images. They use a mapping network consisting of an 8-layer perceptron that aims to map the input latent code to an intermediate latent space. Another popular GAN model is BigGAN [2], a large-scale model trained on ImageNet. Similar to StyleGAN2, it also makes use of intermediate layers by using the latent vector as input, also called *skip-z* inputs, as well as a class vector. Due to its conditional architecture, it can generate images in a variety of categories from ImageNet. In this paper, we work with pre-trained StyleGAN2 and BigGAN models.

2.2. Latent Space Navigation

Recently, several strategies have been proposed to manipulate the latent structure of pre-trained GANs. These methods manipulate images in different ways by editing the latent code and can be divided into two groups.

Supervised Setting. Supervised approaches typically use pre-trained classifiers to guide optimization-based learning to discover interpretable directions that specifically manipulate the properties of interest. InterfaceGAN [26] is a supervised approach that benefits from labeled data including *gender, facial expression* and *age*. It trains binary Support Vector Machines (SVM) [19] on each label and interprets the normal vectors of the obtained hyperplanes as latent directions. GANalyze [6] finds directions for cognitive image properties for a pre-trained BigGAN model using an externally trained *assessor* function. Feedback from the assessor guides the optimization process, and the resulting optimal direction allows manipulation of the desired cognitive attributes. StyleFlow [1] uses attribute-conditioned continuous normalizing flows that use labels to find edit directions in the latent space of GANs.

Unsupervised Setting. One of the unsupervised works proposed by [33] discovers meaningful directions using 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, the method
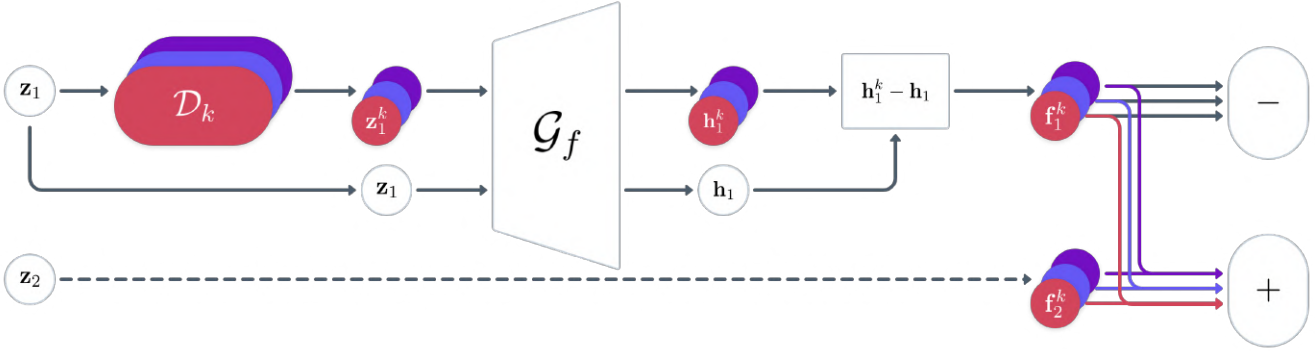
Figure 2: **Illustration of LatentCLR.** First, each latent code is passed through direction models (denoted with $\mathcal{D}_k$) and up to a target feature layer of the GAN (denoted with $\mathcal{G}_f$) to obtain intermediate representations of edited codes. Then, the effects of direction models are computed by subtracting the representation of the original latent code. Finally, pairs produced by the same model are considered as positive, and others as negative, in a contrastive loss.

learns disentangled directions. Ganspace [9] is a sampling based unsupervised method where latent vectors are randomly selected from the intermediate layers of BigGAN and StyleGAN models. Then they propose to use Principal Component Analysis (PCA) [36] to find principal components that are interpreted as semantically meaningful directions. The principal components lead to a variety of useful manipulations, including *zoom, rotation* in BigGAN or changing *gender, hair color, or age* in StyleGAN models. SeFa [27] 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 of the first projection matrix and selecting the eigenvectors with the largest eigenvalues. A different closed-form solution is proposed by [29] that discovers directions without optimization. Another work proposed by [10] exploits task-specific edit functions. They start by applying an editing operation to the original image, e.g., zoom, and minimize the distance between the original image and the edited image to learn a direction that leads to the desired editing operation. [32] provides directions by using a GAN inversion model and aims to change the image in a particular direction without changing the remaining properties. Instead of working on latent codes, [5] uses the space of generator parameters to discover semantically meaningful directions. [18] uses a fixed decoder and trains an encoder to decouple the processes of disentanglement and synthesis. [13] uses post-hoc disentanglement that requires little to no hyperparameters. [16] is a variant of InfoGAN that uses a contrastive regularizer and aims to make the elements of the latent code set clearly identifiable from one another. A concurrent work to ours is proposed by [23] which uses an entropy-based domination loss and a hard negatives flipping strategy to achieve disentanglement.

3. Methodology

In this section, we first introduce preliminaries of contrastive learning and then discuss details of our method.

3.1. Contrastive Learning

Contrastive learning has recently become popular due to leading state-of-the-art results in various unsupervised representation learning tasks. It aims to learn representations by contrasting positive pairs against negative pairs [8] and is used in various computer vision tasks, including data augmentation [3, 20], or diverse scene generation [31]. The core idea of contrastive learning is to move the representations of similar pairs near and dissimilar pairs far.

In this work, we follow a similar approach to SimCLR framework [3] for contrastive learning. SimCLR consists of four main components: a stochastic data augmentation method that generates 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, and a contrastive loss function ℓ that enforces the separation between positive and negative pairs. Given a random mini-batch of N samples, SimCLR generates N positive pairs using the specified 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 [28, 20] over all positive pairs $(\mathbf{x}_i, \mathbf{x}_j)$:

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

where $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}/\|\mathbf{u}\|\|\mathbf{v}\|$ is the cosine similarity function, $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function that

takes the value 1 only when $k = i$, and τ is the temperature parameter. The two networks f and g are trained together. Intuitively, g learns a mapping to a space where cosine similarity represents 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)

Consider a pre-trained GAN, expressed as a mapping function $\mathcal{G} : \mathcal{Z} \to \mathcal{X}$ where $\mathcal{Z}$ is the latent space, usually associated with a prior distribution such as the multivariate Gaussian distribution, 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 look for edit directions $\Delta\mathbf{z}$ such that the image $\mathbf{x}' = \mathcal{G}(\mathbf{z} + \Delta\mathbf{z})$ has semantically meaningful changes with respect to $\mathbf{x}$ while preserving the identity of $\mathbf{x}$. Similar to [9, 27, 33], we limit ourselves to the unsupervised setting, where we aim to identify such edit directions without external supervision, as in [6, 26, 10].

Our intuition is to optimize an *identifiability-based* heuristic, similar to [33], in an intermediate representation space of a pre-trained GAN to find a diverse set of interpretable directions. We search for edit directions $\Delta\mathbf{z}_1, \cdots, \Delta\mathbf{z}_K$, $K > 1$, that have distinguishable effects in the target representation layer. For this end, we calculate *differences* in representations induced by each direction and use a contrastive-learning objective to maximize the identifiability of directions.

More specifically, we generalize *directions* with potentially more expressive conditional mappings called *direction models*. Then, our final approach consist of (i) concurrent directions models that apply edits to the given latent codes, (ii) a target feature layer f of a pre-trained GAN $\mathcal{G}$ that will be used to evaluate direction models, and as well as (iii) a contrastive learning objective for measuring identifiability of each direction model. See Figure 2 for a high-level visualization.

Direction models. The direction model is a mapping $\mathcal{D} : \mathcal{Z} \times \mathbb{R} \to \mathcal{Z}$ that takes latent codes along with a desired edit *magnitude* and outputs edited latent codes, i.e. $\mathcal{D} : (\mathbf{z}, \alpha) \to \mathbf{z} + \Delta\mathbf{z}$, where $\|\Delta\mathbf{z}\| \propto \alpha$. We consider three alternative methods for choosing the direction model: *global, linear* and *non-linear*, which are defined as follows:

- *Global.* We learn a fixed direction θ independent 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, which is a linear dependency.

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

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

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

Note that for all options, we apply ℓ_2 normalization and thus the magnitudes given by α correspond to the direct ℓ_2 distances from the latent code.

The first option, *Global*, is in principle the most limited since it can only find fixed directions and thus edits images without considering the latent code z. However, we note that it is still able to capture common directions such as *zoom, rotation*, or *background removal* on BigGAN. The second option, *Linear* is able to generate *conditional* directions given latent code z, but is still limited for capturing finer-grained directions. The third option is an extension of the *Linear* direction model, where we use a neural network that models the dependency between the direction and the given latent code.

Target feature differences. For each latent code $\mathbf{z}_i, 1 \leq i \leq N$ in the mini-batch of size N, we compute K distinct edited latent codes:

$$\mathbf{z}_i^k = \mathcal{D}(\mathbf{z}_i, \alpha).$$

Then, we calculate corresponding intermediate feature representations,

$$\mathbf{h}_i^k = \mathcal{G}_f(\mathbf{z}_i^k),$$

where $\mathcal{G}_f$ is feed-forward of GAN up to the target layer f. Next, we compute the feature divergences w.r.t. the original latent code,

$$\mathbf{f}_i^k = \mathbf{h}_i^k - \mathcal{G}_f(\mathbf{z_i}).$$

Objective function. For each edited latent code $\mathbf{z}_i^k$, we define the following loss:

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

The intuition behind our objective function is as follows. All feature divergences obtained with the same latent edit operation $1 \leq k \leq K$, i.e., each of $\mathbf{f}_1^k, \mathbf{f}_2^k, \cdots \mathbf{f}_N^k$, are considered as *positive* pairs and contribute to the numerator. All other pairs obtained with a different edit operation, e.g., $\mathbf{f}_1^k \neq \mathbf{f}_N^l$, $l \neq k$, are considered as *negative* pairs and

Figure 3: (a) Directions for general image editing operations such as *zoom*, or *rotation* discovered from the ImageNet Bulbul class, where we shift the latent code in a particular direction with increasing or decreasing α. (b) Transferred directions from the *Bulbul* class to various other ImageNet classes.

contribute to the denominator. This can be viewed as a generalization of the NT-Xent loss (Eq. 1), where we have N-tuples of groups, one for each direction model. With this generalized contrastive loss, we enforce latent edit operations to have orthogonal effects on the features.

Utilizing Layer-wise Styles. Ganspace [9] discovered that the layer-wise structure of StyleGAN2 and BigGAN models can be used for fine-grained editing. By applying their directions only to a limited set of layers in test-time, they achieve less entanglement in editing and superior granularity. SeFa [27] can find more detailed directions by concatenating weight matrices and identifying eigenvectors. In contrast to Ganspace, our method can consider such layer-wise structure in training time. And, in contrast to SeFa, our method, additionally, can fuse the effects of all selected layers into the target feature layer due to its flexible optimization-based objective.

4. Experiments

We evaluate the proposed method for detecting semantically meaningful directions using different models and datasets. We apply the proposed model to BigGAN and StyleGAN2 on a wide range of datasets, including human faces (FFHQ) [11], LSUN Cats, Cars, Bedrooms, Church and Horse datasets [37] and ImageNet. We also compare our method with state-of-the-art unsupervised methods

[9, 27] and conduct several qualitative and quantitative experiments to demonstrate the effectiveness of our approach. Next, we discuss our experimental setup and then present results on BigGAN and StyleGAN2 models.

4.1. Experimental Setup

For BigGAN experiments, we choose *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 includes 100k iterations), which takes about 20 minutes. For StyleGAN2 experiments, we choose *batch size* = 8, $K = 100$ directions, *truncation* = 0.7, *feature layer* = *conv1* and train StyleGAN2 models for 5 epochs (each epoch corresponds to 10k iterations), which takes about 12 minutes. We use 1-3 dense layers consisting of units corresponding to latent space (128, 256, or 512), with ReLU and batch normalizations. For our experiments, we use PyTorch framework and two NVIDIA Titan RTX GPUs.

Choice of K. To learn K 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 made by [33]. For BigGAN, we used $K = 32$ directions since the latent space is 128-dimensional and most interpretable directions such as *zoom*, *rotation* or *translation* can be obtained with a relatively small number
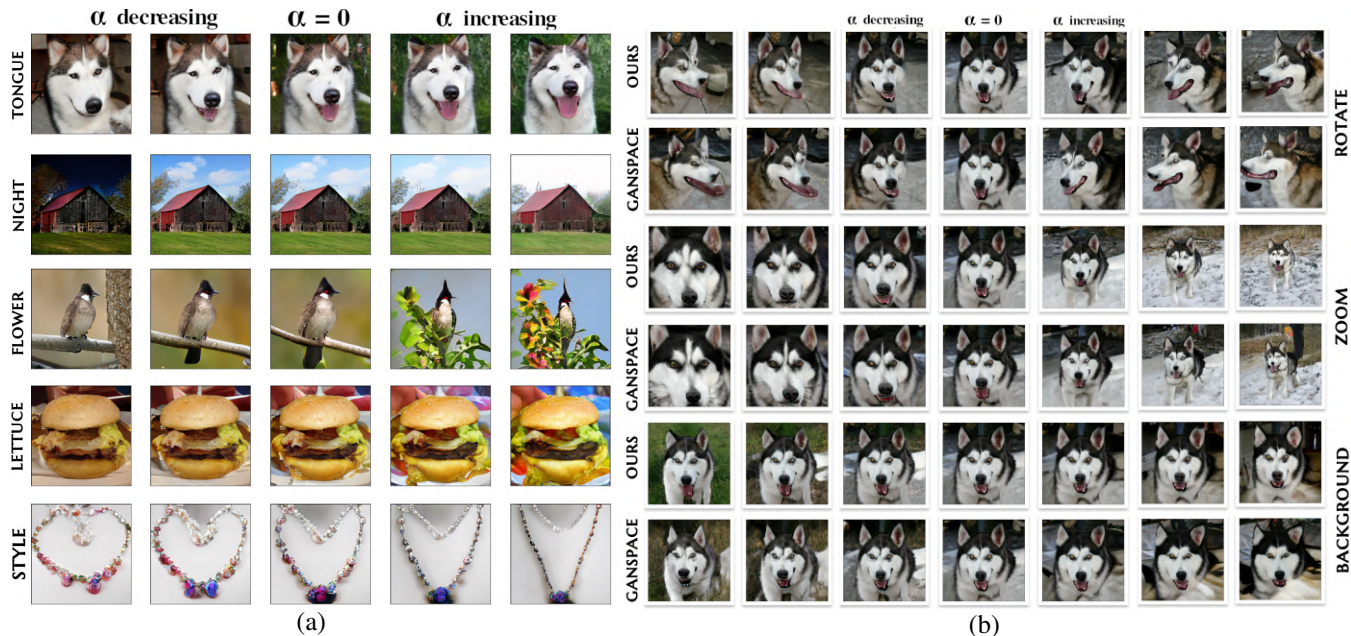
Figure 4: (a) Class-specific directions discovered by our method in several ImageNet classes on the BigGAN model. (b) A comparison of the directions *rotate, zoom* and *background change* between our method and Ganspace.

of directions. For StyleGAN2, we used $K = 100$ directions since the latent space is 512-dimensional.

Layers. To avoid any bias between the competing methods Ganspace and SeFa, we use the same set of StyleGAN2 layers that comes from Ganspace. We also note that slight differences in re-scoring analysis or visuals might be caused due to applying different magnitudes of change across methods, as they are not directly convertible between methods. To minimize this effect, we use the same sigma values in Ganspace as specified for each direction in their public repository, and $\{-3, +3\}$ for SeFa as provided in the official implementation.

4.2. Results on BigGAN

We evaluate our approach using a pre-trained BigGAN model conditionally trained on 1000 ImageNet classes. We trained our model on an arbitrary class, *Bulbul*, and obtained $K = 32$ directions.

Qualitative Results. Our visual analysis shows that our model is able to detect several semantically meaningful directions such as *zoom, rotation, contrast* as well as some finer-grained features such as *background removal, sitting*, or *green background* (see Figure 1 and Figure 3 (a)). As can be seen from the figures, our method is able to manipulate the original image (labelled α=0) by shifting the latent

code towards the interpretable direction (increasing α) or backwards (decreasing α).

Transferability of directions. After verifying that the directions obtained for the class *Bulbul* are capable of manipulating the latent codes for multiple semantic directions, we investigated how transferable the discovered directions are to other ImageNet classes.

Our visual analysis shows that the directions learned from the *Bulbul* class are applicable to a variety of ImageNet classes and are able to zoom a *Trenchcoat*, rotate a *Goose*, apply contrast to *Volcano*, add greenness to *Castle* classes (see Figure 3 (b)) as well as removing the background from a *Teapot* object (see Figure 1). An interesting transferred direction is *Sitting* (see Figure 3 (a)), which manipulates the latent code so that the bird in the Bulbul class sits on a tree branch when we increase α. We find that this direction, when applied to the class *Bulldog*, also causes the dog to stand up or sit down depending on how we move in the positive or negative direction (see Figure 3 (b)).

Diversity of the directions. Next, we investigate whether we can discover unique directions when we train our model on different ImageNet classes. Figure 4 (a) shows some directions discovered by our model, such as adding *tongue* on *Husky* class, changing the *time of day* in *Barn* class, adding *flowers* in *Bulbul* class, adding *lettuce* on *Cheeseburger* class, or changing the *shape* in *Necklace* class.
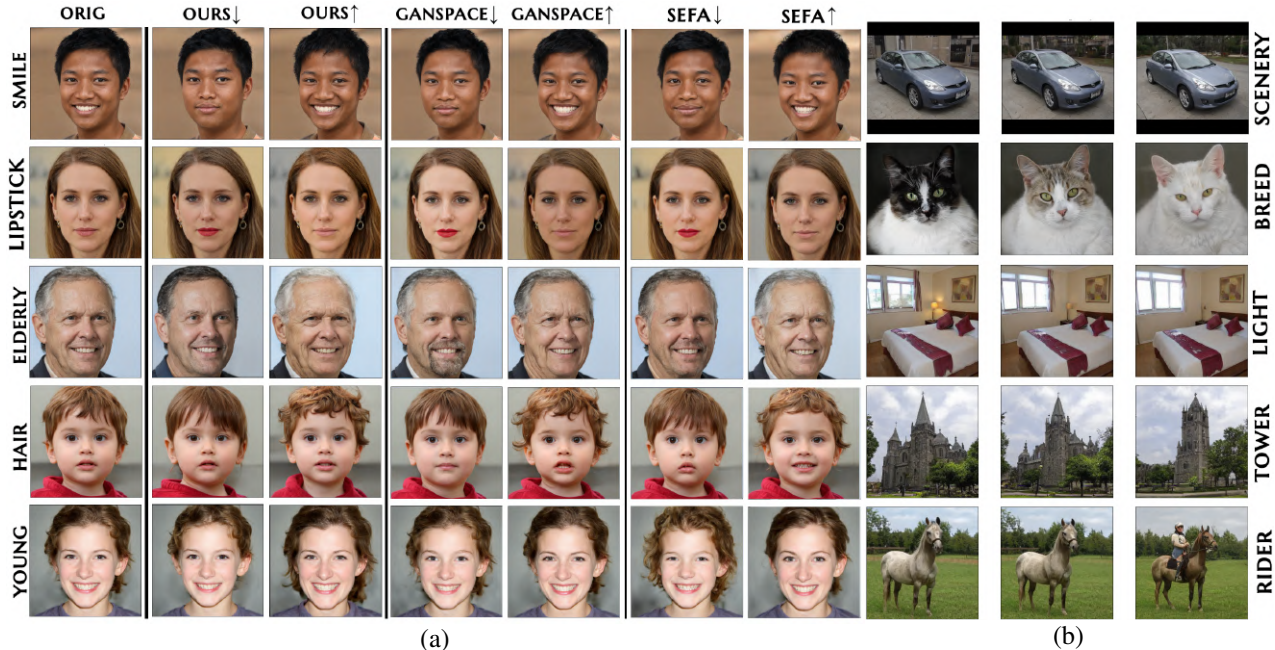
Figure 5: (a) Comparison of manipulation results on FFHQ dataset with Ganspace and SeFa methods. The leftmost image represents the original image, while images denoted with ↑ and ↓ represent the edited image moved in the positive or negative direction. (b) Directions discovered by our method in various LSUN [37] datasets.
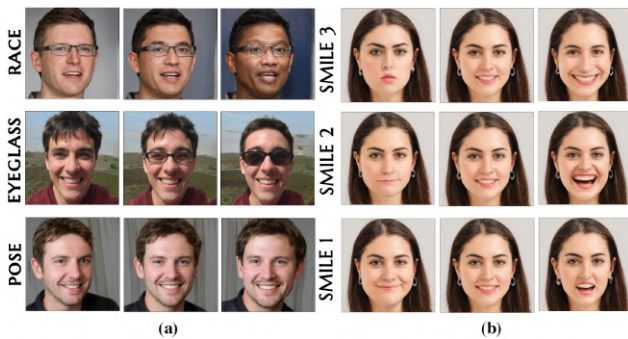

Figure 6: Additional directions discovered by our method in FFHQ dataset with StyleGAN2 (left). Different types of *smile* directions (denoted with *SMILE 1-3*) discovered by the non-linear approach (right).

Comparison with other methods. We visually compare the directions obtained with our method to Ganspace using the *Husky* class (see Figure 4 (b)). For each direction in Ganspace, we use the parameters given in the open-source implementation*. For both methods, we use the same initial image (represented by $\alpha = 0$) and move the latent code towards the direction (as α increases) and far away from the direction (as α decreases). We used the original α settings (i.e., the *sigma* setting) for each direction as provided in

---
*https://github.com/harskish/ganspace

the Ganspace implementation to avoid any bias that may be caused by tuning the parameter. We find that both methods achieve similar manipulations for the *zoom, rotation*, and *background change* directions, while our method causes less entanglement. For example, we note that Ganspace tends to increase *tongue* with *rotation* or add background objects with increasing *zoom*.

User Study. To understand how the directions found by our method match human perception, we conduct a 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 images on BigGAN's *Bulbul* class. Participants are shown the original image in the center and $-\alpha$ and $+\alpha$ values on the left and right sides, respectively. Following the same approach as [27], 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 17.43 on average with "Yes", 7.43 with "Maybe," and 5.75 with "No". For *Question 2* we got *Yes* on average 14.37 directions, *Maybe* on average

10.03 directions, and *No* on average 6.21 directions. These results indicate that out of $K = 32$ directions, participants found 82% semantically meaningful and 80.59% directions containing an obvious content change.

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: Re-scoring Analysis on Ganspace, SeFA and our method where we compare *Smile, Age* and *Lipstick* attributes using FFHQ dataset on StyleGAN2.

4.3. Results on StyleGAN2

We apply our method on StyleGAN2 to a wide range of datasets and compare our results with state-of-the-art unsupervised methods Ganspace and SeFa.

Qualitative Results. First, we visually examined the directions found by our method in several datasets, including FFHQ, and LSUN Cars, Cats, Bedroom, Church, and Horse datasets (see Figure 1 and Figure 5 (b)). Our method is capable of discovering 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* in LSUN Horse datasets.

Comparison with other methods. We compare how the directions found on FFHQ differ across methods. Figure 5 (a) shows the visual comparison between several directions found in common by all methods, including the directions *Smile, Lipstick, Elderly, Curly Hair* , and *Young*. As can be seen from the figures, all methods perform similarly and are able to manipulate the images towards the desired attributes. Figure 6 (a) illustrates various directions including *Race, Eyeglass*, and *Pose*. Figure 6 (b) illustrates three different *smile* directions discovered by our non-linear model.

Re-scoring analysis. To understand how our method compares to the competitors in quantitative terms, we performed a re-scoring analysis after [27] using attribute predictors to understand whether the manipulations changed the images towards the desired attributes. We used attribute predictors released by StyleGAN2 for the directions *Smile*

and *Lipstick*, as these are the only two attributes for which a predictor is available, and which are simultaneously found in common by all methods. For the *Age* direction, we used an off-the-shelf age predictor [25]. Table 1 shows our results for the re-scoring analysis for three attributes: *Age*, *Lipstick*, and *Smile* for negative (labelled ↓) and positive (labelled ↑) directions. To see how the scores change, 500 images are randomly generated for each property. The average score of the predictors for *Lipstick* property is 0.43 ±0.5, the average score for *Age* is 0.27 ±0.1, and the average score for *Smile* is 0.74 ±0.43. Moving the latent codes in the negative direction, we find that our method, as well as the Ganspace and SeFa methods, decrease scores in similar ranges on the *Age* and *Lipstick* properties. We find that both our method and Ganspace are able to significantly reduce the score (from 0.74 to 0.11 and 0.05, respectively) when moving towards the negative direction for the *Smile* property, while SeFa reduces the score from 0.74 to 0.50 on average. When we move the latent codes in the positive direction, all methods achieve comparable results.

5. Limitations

Our method uses a pre-trained GAN model as input, so it is limited to manipulating GAN-generated images only. However, it can be extended to real images by using GAN inversion methods [40] by encoding the real images into the latent space. Like any image synthesis tool, our method also poses similar concerns and dangers in terms of misuse, as it can be applied to images of people or faces for malicious purposes, as discussed in [14].

6. Conclusion

In this study, we propose a framework that uses contrastive learning to learn unsupervised directions. Instead of discovering fixed directions such as [33, 9, 27], our method can discover non-linear directions in pre-trained StyleGAN2 and BigGAN models and leads to multiple distinct and semantically meaningful directions that are highly transferable. We demonstrate the effectiveness of our approach on a variety of models and datasets, and compare it to state-of-the-art unsupervised methods. We make our implementation available at https://github.com/catlab-team/latentclr.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021.

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

[3] 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.

[4] 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.

[5] 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*, pages 3671–3680, 2021.

[6] 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.

[7] 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.

[8] 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.

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

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

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

[12] 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.

[13] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. On disentangled representations extracted from pretrained gans. 2020.

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

[15] 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.

[16] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning*, pages 6127–6139. PMLR, 2020.

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

[18] Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. Face identity disentanglement via latent space mapping. *arXiv preprint arXiv:2005.07728*, 2020.

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

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

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

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

[23] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Do generative models know disentanglement? contrastive learning is all you need. *arXiv preprint arXiv:2102.10543*, 2021.

[24] 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.

[25] 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.

[26] 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.

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

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

[29] Nurit Spingarn-Eliezer, Ron Banner, and Tomer Michaeli. Gan" steerability" without optimization. *arXiv preprint arXiv:2012.05328*, 2020.

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

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

[32] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1710.10916, 2017.

[39] 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.

[40] 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.