
Closed-Form Factorization of Latent Semantics in GANs

Yujun Shen, Bolei Zhou

The Chinese University of Hong Kong
 {sy116, bzhou}@ie.cuhk.edu.hk

Abstract

A rich set of semantic attributes has been shown to emerge in the latent space of the Generative Adversarial Networks (GANs) trained for synthesizing images. In order to identify such latent semantics for image manipulation, previous methods annotate a collection of synthesized samples and then train supervised classifiers in the latent space. However, they require a clear definition of the target attribute as well as the corresponding manual annotations, severely limiting their applications in practice. In this work, we examine the internal representation learned by GANs to reveal the underlying variation factors in an *unsupervised* manner. By studying the essential role of the fully-connected layer that takes the latent code into the generator of GANs, we propose a general *closed-form* factorization method for latent semantic discovery. The properties of the identified semantics are further analyzed both theoretically and empirically. With its fast and efficient implementation, our approach is capable of not only finding latent semantics as accurately as the state-of-the-art supervised methods, but also resulting in far more versatile semantic classes across multiple GAN models trained on a wide range of datasets.¹

1 Introduction

Generative Adversarial Networks (GANs) [8] have achieved tremendous success in image synthesis. Recent work [7, 14, 23, 21, 25] has found that when learning to synthesize images, GANs spontaneously represent multiple interpretable attributes in the latent space, such as gender for face synthesis [23] and lighting condition for scene synthesis [25]. By properly identifying these semantics, we can reuse the knowledge learned by GANs to reasonably control the image generation process, enabling a wide range of editing applications, such as face manipulation [23, 9] and scene editing [25, 27].

The crux of interpreting the latent space of GANs is to find the meaningful subspaces corresponding to the human-understandable attributes [7, 14, 23, 21, 25]. Through that, moving the latent code towards the direction of a certain subspace can accordingly change the semantic occurring in the synthesized image. However, due to the high dimensionality of the latent space as well as the large diversity of image semantics, finding valid directions in the latent space is extremely challenging. Existing supervised learning-based approaches typically first randomly sample a large amount of latent codes, then synthesize a collection of images and annotate them with some pre-defined labels, and finally use these labeled samples to learn a separation boundary in the latent space. To get the labels for training the boundary, they either introduce pre-trained semantic predictors [7, 23] or utilize some simple statistical information of the image (*e.g.*, object position and color tone) [14, 21].

Several critical limitations rise from the previous semantic discovery methods which heavily depend on the pre-defined semantics and the preparation of annotated samples. On one hand, relying on

¹Code and demo video can be found at <https://genforce.github.io/sefa/>.

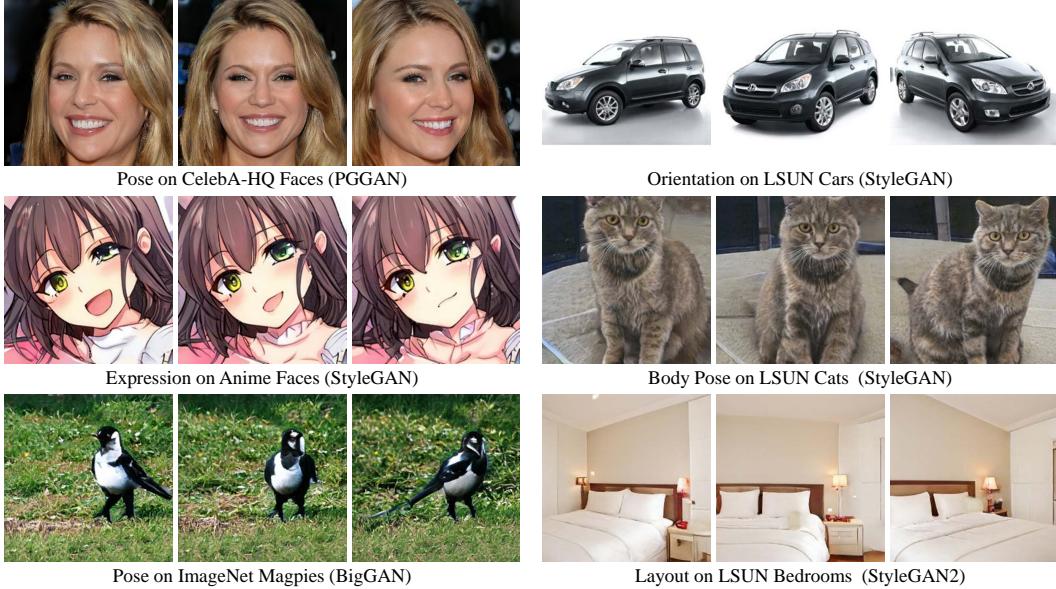


Figure 1: Versatile semantics unsupervisedly found from different types of GAN models trained on diverse datasets, including PGGAN [15], StyleGAN [16], BigGAN [4], and StyleGAN2 [17]. For each set of images, the middle one is the original sample, while the left and the right are the samples by moving the latent code toward and backward the interpretable direction found by **SeFa**.

pre-defined semantics will hinder the algorithm from being applied to the case when the semantic predictors are missing. On the other hand, sampling is unstable such that a different collection of synthesis may lead to a different boundary search. Besides, to identify some rare attributes, a considerably large amount of samples have to be generated to ensure harvesting enough positive examples for the boundary training, which can be quite time-consuming and biased. Therefore, instead of utilizing the synthesized samples as an intermediate step, we directly look into the generation mechanism of GANs to interpret its internal representation. More concretely, for all the GAN architectures based on neural networks, a fully-connected layer is always employed as the very first step to take the input latent code into the generator. It provides the driving force for projecting the latent space to a transformed space. Such transformation actually filters out some negligible directions in the latent space and highlights the directions that are critical for image synthesis. If we can identify these important latent directions, we are able to control the image generation process, *i.e.*, editing the semantics of the synthesized image.

To this end, we provide a novel and simple *closed-form* approach, termed as **SeFa**, for latent Semantic Factorization in GANs. Different from the three-steps pipeline (*i.e.*, sampling, labelling, and boundary searching) which is commonly used in existing methods, our algorithm *only* employs the learned weight of the GAN model for semantic discovery. Extensive experiments suggest that our approach is able to identify versatile latent semantics with an extremely fast and efficient implementation (*i.e.*, *less than 1 second*). Fig. 1 shows some manipulation examples where we can rotate the object in the image even without knowing its underlying 3D model or pose label. Compared to the existing supervised methods and some very recent unsupervised baselines, our approach is able to identify interpretable dimensions more accurately and in a wider range. Furthermore, our approach is general and flexible such that it can well support finding human-understandable semantics emerging across multiple GAN models (*i.e.*, PGGAN [15], StyleGAN [16], BigGAN [4], and StyleGAN2 [17]) that are trained on a wide range of datasets, as shown in Fig. 1.

1.1 Related Work

Generative Adversarial Networks. GAN [8] has significantly advanced image synthesis in recent years [22, 2, 15, 4, 16, 17]. The generator in GANs is able to take a randomly sampled latent code as the input and output a high-fidelity image. Existing GAN models are commonly built on deep convolutional neural networks [22]. To take the latent code into the generator, a fully-connected layer is employed in mapping the latent code to an initial feature map from which the convolutional layers

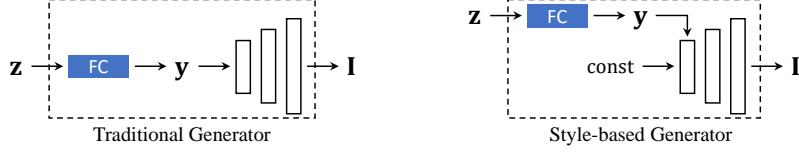


Figure 2: Two different types of generator. The traditional generator [15] uses the fully-connected layer (FC) to map the latent code to the initial feature map for convolutional layers. The style-based generator [16] uses FC to map the latent code to layer-wise style codes, which modulate the feature maps of each convolutional layer. Each layer has its own FC, but we only show one layer for simplicity.

start [22, 2, 15]. Recently, this idea is improved by the style-based generator [16, 17], where the latent code is mapped to layer-wise style codes and then fed into each convolutional layer through Adaptive Instance Normalization (AdaIN) [13]. In this case, the transformation of the fully-connected layer as the style mapping is still the very first step for the generator to process the latent code.

Latent Semantic Interpretation. Generative models show a great potential in learning variation factors from the data in an unsupervised manner. Chen et al. [5] and Higgins et al. [12] propose to add regularizer into the training process to explicitly learn an interpretable factorized representation, but the number of factors should be pre-defined before training and it remains unknown if they are applicable to recent high-fidelity image synthesis models, like StyleGAN [16]. Recent work has found that unconditional GANs automatically encode various semantics in the intermediate feature maps [3] and the initial latent space [7, 14, 23, 25]. By properly identifying these semantics, we are able to faithfully control the image generation process. However, existing methods require preparing a collection of synthesized images, assigning these images with semantic scores, and then using the image-score pairs for supervised semantic search, which are severely limited by available semantic predictors. Some concurrent work studies unsupervised semantic discovery in GANs. Voynov and Babenko [24] introduces the regularizer proposed by InfoGAN [5] into the semantic search process but requires a pre-defined number of semantics. Härkönen et al. [10] proposes to skip the labelling process and perform PCA on the sampled data to find primary directions in the latent space. However, both of them are still based on sampling a large amount of data. Differently we propose a *closed-form* factorization method for latent semantic interpretation, which is independent of sampling and also does not require any prior supervised information such as semantic labels or classifiers.

2 Methodology

2.1 Problem Statement

The generator $G(\cdot)$ in GANs learns the mapping from the d -dimensional latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ to a higher dimensional image space $\mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$, as $\mathbf{I} = G(\mathbf{z})$. Here, $\mathbf{z} \in \mathcal{Z}$ and $\mathbf{I} \in \mathcal{I}$ denote the input latent code and the output image respectively. There are two common ways to feed a vector \mathbf{z} into the generator, as shown in Fig.2. One is to map \mathbf{z} to a feature map from which the convolutional layers start [22, 2, 15], while the other is to map \mathbf{z} to layer-wise style codes which control the output feature map of each convolutional layer through AdaIN [16, 17]. Both of these approaches require a linear projection implemented by the fully-connected layer (FC) to do the mapping. Hence, we reformulate the generation process as

$$\mathbf{I} = G'(FC(\mathbf{z})) \triangleq G'(\mathbf{y}), \quad (1)$$

where $G'(\cdot)$ represents the remaining part of the generator except the learned FC at the beginning.

The latent space of GANs has been shown to encode rich semantic knowledge [7, 14, 23, 25] with vector arithmetic property [22]. Such vector arithmetic property is often realized by moving the latent code towards some certain direction as

$$\mathbf{z}' = \mathbf{z} + \alpha \mathbf{n}, \quad (2)$$

where $\mathbf{n} \in \mathbb{R}^d$ denotes the direction corresponding to a particular attribute and α is the moving step. After that, the semantic occurring in the output image $\mathbf{I} = G'(FC(\mathbf{z}'))$ will vary accordingly. In this work, we aim at finding the semantically meaningful directions \mathbf{n} in an unsupervised manner.

2.2 Unsupervised Semantic Factorization

From Eq.(2), we observe that the semantic is actually determined by the latent direction \mathbf{n} , which is independent of the sampled code \mathbf{z} . In other words, we would like to find \mathbf{n} , instead of \mathbf{z} , that can cause the shift of \mathbf{I} to the most extent. According to Eq.(1), when $G'(\cdot)$ is fixed, the shift of \mathbf{I} hinges on the change of \mathbf{y} . Here, we make an *assumption* that a large change of \mathbf{y} will lead to a large content variation of \mathbf{I} . From this perspective, our goal turns into finding the directions \mathbf{n} that can cause the significant change of \mathbf{y} .

To better model the relationship between \mathbf{n} and \mathbf{y} , we revisit the linear projection implemented by the fully-connect layer that connects the latent code and the generator. Let $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^m$ denote the weight and bias respectively, where m is the dimension of the projected space. Then, we have $\mathbf{y} = FC(\mathbf{z}) = \mathbf{Az} + \mathbf{b}$. After modulating the latent code with Eq.(2), we can compute the change of the projected code as

$$\Delta\mathbf{y} = FC(\mathbf{z}') - FC(\mathbf{z}) = (\mathbf{A}(\mathbf{z} + \alpha\mathbf{n}) + \mathbf{b}) - (\mathbf{Az} + \mathbf{b}) = \alpha\mathbf{An}. \quad (3)$$

From Eq.(3), we can tell that, to make a large change of \mathbf{y} , we need to find a direction such that this direction brings a large norm after the ‘‘projection’’ by \mathbf{A} . Intuitively, this transformation in the fully-connected layer (\mathbf{A}) plays the essential role of ‘‘semantic selector’’. For whatever semantic (\mathbf{n}) encoded in the latent space, it has to go through this selector to be reflected in the final synthesis (\mathbf{I}). Therefore, we factorize this projection as the guidance to explore the latent semantics by solving the following optimization problem

$$\mathbf{n}^* = \arg \max_{\{\mathbf{n} \in \mathbb{R}^d : \mathbf{n}^T \mathbf{n} = 1\}} \|\mathbf{An}\|_2^2, \quad (4)$$

where $\|\cdot\|_2$ denotes the l_2 norm. Here, we set $\alpha = 1$ and use unit vector for all directions to make sure they are comparable.

When the case comes to finding k most important semantics $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$, Eq.(4) turns into

$$\mathbf{N}^* = \arg \max_{\{\mathbf{N} \in \mathbb{R}^{d \times k} : \mathbf{n}_i^T \mathbf{n}_i = 1 \forall i=1, \dots, k\}} \sum_{i=1}^k \|\mathbf{An}_i\|_2^2, \quad (5)$$

where $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k]$ denotes the top- k semantics.

To solve this problem, we introduce the Lagrange Multiplier as

$$\begin{aligned} \mathbf{N}^* &= \arg \max_{\mathbf{N} \in \mathbb{R}^{d \times k}} \sum_{i=1}^k \|\mathbf{An}_i\|_2^2 - \sum_{i=1}^k \lambda_i (\mathbf{n}_i^T \mathbf{n}_i - 1) \\ &= \arg \max_{\mathbf{N} \in \mathbb{R}^{d \times k}} \sum_{i=1}^k (\mathbf{n}_i^T \mathbf{A}^T \mathbf{A} \mathbf{n}_i - \lambda_i \mathbf{n}_i^T \mathbf{n}_i + \lambda_i). \end{aligned} \quad (6)$$

By taking the partial derivative on each \mathbf{n}_i , we have

$$2\mathbf{A}^T \mathbf{A} \mathbf{n}_i - 2\lambda_i \mathbf{n}_i = 0. \quad (7)$$

From Eq.(7) we can see that all possible solutions should be the eigenvectors of the matrix $\mathbf{A}^T \mathbf{A}$. Hence, to get the maximum objective value and also make $\{\mathbf{n}_i\}_{i=1}^k$ to be distinguishable from each other, we choose columns of \mathbf{N} as the eigenvectors of $\mathbf{A}^T \mathbf{A}$ associated with the k highest eigenvalues.

2.3 Property of the Discovered Semantics

In this part we discuss about some important properties of the discovered semantics. As mentioned in Sec.2.2, all semantic directions are the eigenvectors of matrix $\mathbf{A}^T \mathbf{A}$, which is positive semi-definite. Hence, we always have the eigen decomposition

$$\mathbf{A}^T \mathbf{A} = \mathbf{Q} \Lambda \mathbf{Q}^T, \quad (8)$$

where Λ is a diagonal matrix, indicating the eigenvalues, while \mathbf{Q} is an orthogonal matrix, containing all the eigenvectors.

Table 1: Comparison between the semantics identified by different methods, including performing PCA on a collection of sampled latent codes [10] and our *closed-form* solution. Semantics learned by InterFaceGAN [23] are used as the “ground-truth”. “Dist.” indicates the cosine distance (smaller is better), while “No.” denotes the 0-based index of the eigenvector (smaller means primary direction).

		Pose		Gender		Age		Eyeglasses		Smile	
		Dist.	No.								
PGGAN \mathcal{Z} Space	PCA	0.90	23	0.83	1	0.88	1	0.86	1	0.86	10
	Ours	0.04	2	0.16	1	0.45	4	0.52	1	0.69	20
StyleGAN \mathcal{W} Space	PCA	0.28	9	0.71	2	0.73	5	0.72	5	0.73	25
	Ours	0.13	1	0.64	3	0.80	6	0.65	6	0.65	37

Obviously, each \mathbf{n}_i is a column of \mathbf{Q} . We therefore have $\mathbf{N}^T \mathbf{N} = \mathbf{I}_k$, where \mathbf{I}_k represents the k -dimensional identity matrix. It means that all semantic directions found by our algorithm are orthogonal to each other in the latent space. We also have

$$\Delta \mathbf{y}_i^T \Delta \mathbf{y}_j = \mathbf{n}_i^T \mathbf{A}^T \mathbf{A} \mathbf{n}_j = \mathbf{n}_i^T (\lambda_j \mathbf{n}_j) = 0 \quad \forall i \neq j, \quad (9)$$

which means that the variations of the outputs of FC derived by different directions are also orthogonal to each other. Based on these two observation, we can expect the semantics corresponding to $\{\mathbf{n}_i\}_{i=1}^k$ are disentangled from each other.

3 Experiments

In this section, we empirically evaluate the effectiveness of the proposed approach. Before going into details, we first introduce the models and datasets used in this work and the implementation details.

Models and Datasets. We conduct extensive experiments on the state-of-the-art GAN models, including PGGAN [15], StyleGAN [16], BigGAN [4], and StyleGAN2 [17]. They are trained on diverse datasets, including human faces (CelebA-HQ [15] and FF-HQ [16]), anime faces [1], scenes and objects (LSUN [26]), streetscapes [19], and ImageNet [6]. For quantitative analysis on faces, we train an attribute predictor on CelebA dataset [18] with ResNet-50 [11], following prior work [23].

Implementation Details. For PGGAN, we apply our algorithm onto the very first FC layer. For StyleGAN and StyleGAN2, which are with the style-based generator, we choose the style mapping layers of each convolutional block. Our algorithm is flexible to interpreting all or any particular layers. Concretely, we concatenate the weight matrices from all target layers along the first axis, resulting in a larger matrix. Note that, besides the latent space \mathcal{Z} , StyleGAN introduces a more disentangled space \mathcal{W} [16]. We do experiments on \mathcal{W} space for StyleGAN and StyleGAN2 since $\mathbf{w} \in \mathcal{W}$, instead of $\mathbf{z} \in \mathcal{Z}$, is the code fed into the generator. For BigGAN, which maps the latent code to both the initial feature map and the layer-wise styles, we choose to combine the weights from all these fully-connected layers together. Since BigGAN employs an embedding vector, which is extended to the latent code, for conditional synthesis, we only use the weight parameters that are associated with the latent code part (*i.e.*, the first half columns). Before performing eigen decomposition, we normalize each row of the weight matrix \mathbf{A} since we focus more on the direction than the distance.

3.1 Comparison with Unsupervised Baselines

Post-Annotation of the Discovered Semantics. After getting the potential semantic directions, we need to align them with human perception by assigning them with interpretable meanings. Here, we use the semantic directions found by existing supervised method, *i.e.*, InterFaceGAN [23], as the “ground-truth”. For a particular attribute, we compare all eigen directions from our algorithm with the “ground-truth” direction and choose the one with the smallest cosine distance. Tab. 1 shows the results, where we can see that pose, age, and gender all emerge as the primary directions using our method on both the traditional generator (PGGAN [15]) and the style-based generator (StyleGAN [16]). The smiling semantic does not cause obvious change of the image (*i.e.*, only the mouth part varies), and hence is not with high eigenvalues. But our method can still spot this variation.

Comparison with Sampling-based Unsupervised Baseline. We then compare our method with the sampling-based unsupervised baseline, which is to first sample a collection of latent codes and then

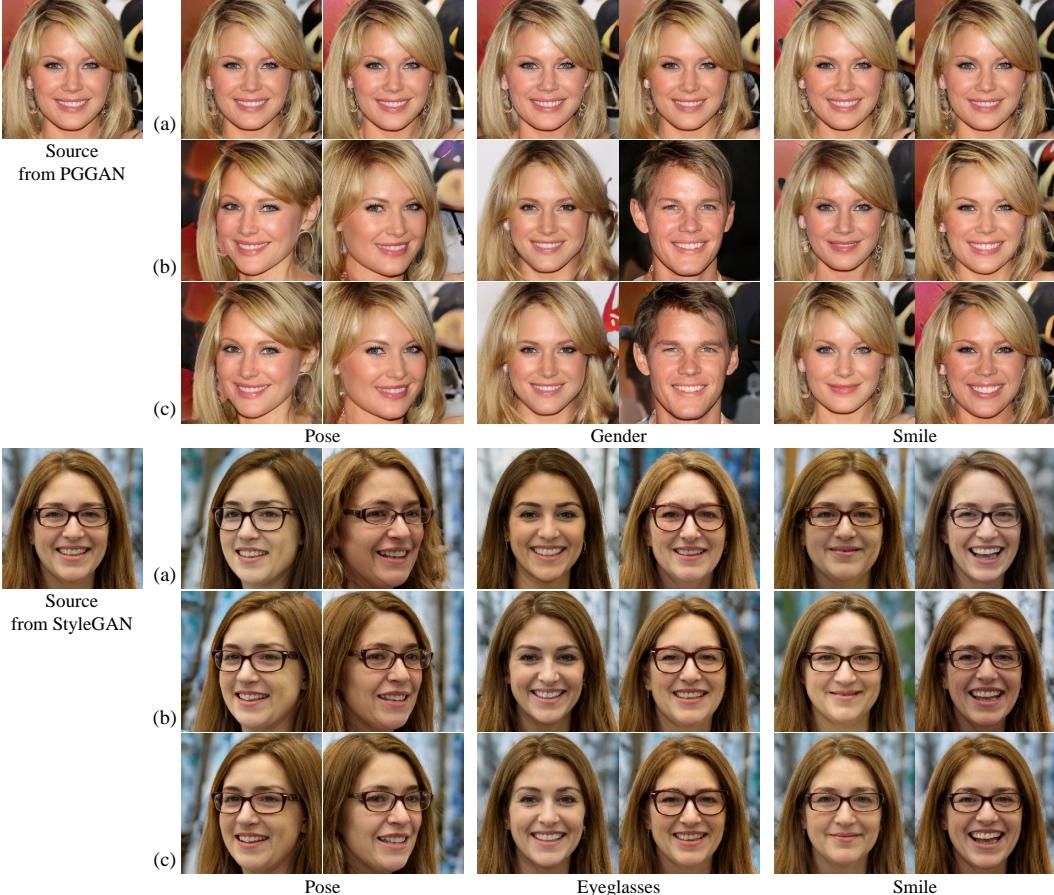


Figure 3: Qualitative comparison of the semantics found by (a) sampling-based unsupervised approach [10], (b) our *closed-form* solution, **SeFa**, and (c) the supervised method, InterFaceGAN [23].

perform PCA on these samples to find principle directions [10]. Here, we use $500K$ samples for PCA. From Tab.1, we can tell that the sampling-based method would fail on the traditional generator since the latent codes are simply subject to a normal distribution. Conducting PCA on the normally distributed data cannot give meaningful directions. Meanwhile, our method also achieves better results on the \mathcal{W} space of StyleGAN. For example, pose is identified as the 9-th direction by PCA but as the 1-st direction by our algorithm. Also, directions from our method show a smaller distance to the “ground-truth”. It is worth to mention that our method does not rely on the sampled data and hence is more stable. We also show some qualitative results in Fig.3. We can see that our results (row (b)) are more close to those achieved by the supervised approach InterFaceGAN (row (c)). For example, the identity and hair style change when editing pose using PCA on StyleGAN (row (a)).

Comparison with Learning-based Unsupervised Baseline. InfoGAN [5] proposed to explicitly learn a factorized representation in an unsupervised manner. We compare our method with the PGGAN model trained by adding the regularizer to maximize the mutual information [20], which is shown in Fig.4. We have two main advantages. First, InfoGAN requires knowing the number of factors before the training process. If we want to add more semantics, the model should be re-trained from the scratch. Second, our algorithm identifies the semantics automatically learned by GANs, which are more accurate than the semantics learned with the additional regularizer. Taking pose manipulation in Fig.4 as an example, the hair color varies when using Info-PGGAN for editing.

3.2 Comparison with Supervised Approach & Semantic Property Analysis

We compare our algorithm with the state-of-the-art supervised method for latent semantic discovery, InterFaceGAN [23], from the perspectives of (a) how different semantics are disentangled from each other, and (b) how diverse the identified semantics are.

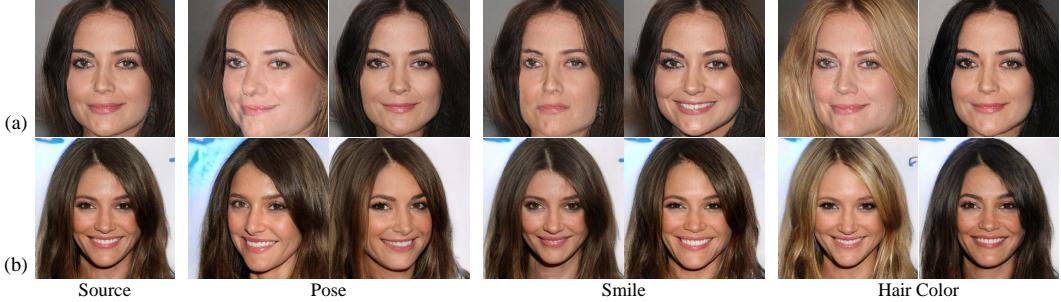


Figure 4: Qualitative comparison of the semantics found by (a) Info-PGGAN [20, 5] and (b) SeFa.

Table 2: Re-scoring analysis on different methods by evaluating how the semantic scores change via modulating the latent codes. Each row shows the results by moving the latent codes towards a certain direction.

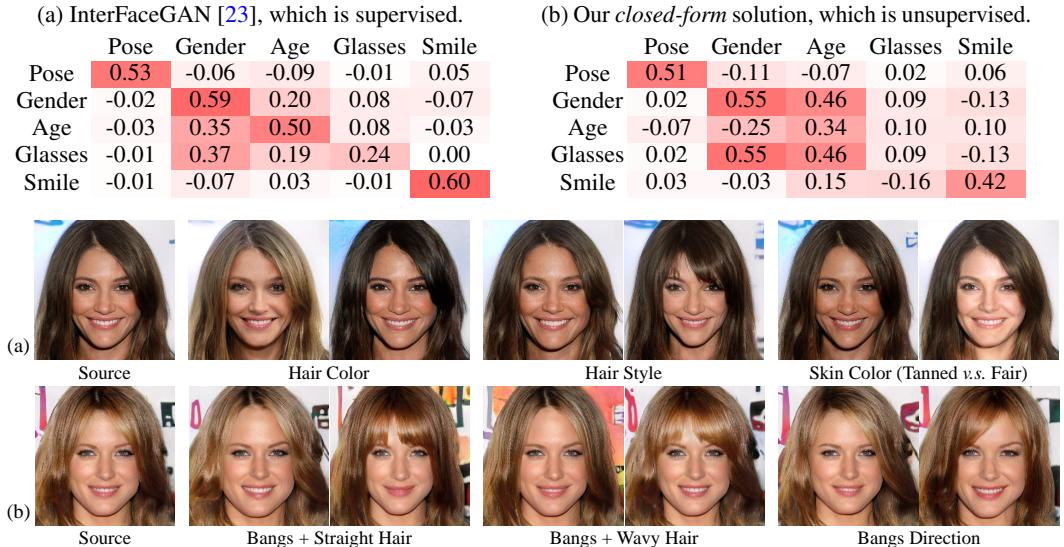


Figure 5: (a) Diverse semantics, which can *not* be identified by InterFaceGAN [23] due to the lack of semantic predictors. (b) Diverse hair styles, which can *not* be described as a binary attribute.

Disentanglement Comparison. We perform re-scoring analysis to quantitatively evaluate the disentanglement between different semantics. In particular, we first randomly sample 2,000 images, then manipulate them along a certain semantic direction, and finally use pre-trained attribute predictors to check how the scores corresponding to different attributes vary in the manipulation process. Tab.2 shows the comparison results. We can see that our unsupervisedly discovered semantics show similar disentanglement property as the supervised method. For example, pose and smile are almost independent from the other three attributes, while gender, age, glasses are highly entangled with each other. An interesting observation is that when manipulating age and gender with InterFaceGAN (Tab.2(a)), the gender score always increases. However, when using our approach (Tab.2(b)), the gender score increases when manipulating gender yet decreases when manipulating age. That is because even our algorithm cannot perfectly disentangle age from gender (this may be caused by the data bias [23]), the boundaries themselves are disentangled, as proved in Sec.2.3. For example, “masculinizing + aging” and “feminizing + aging” are also two orthogonal directions in the latent space. Furthermore, the labels of the directions identified by our method are assigned using InterFaceGAN (see Sec.3.1), which are not 100% accurate. Like, gender and glasses are assigned to a same direction. If we manually label these directions, it may lead to a better disentanglement.

Diversity Comparison. Supervised approach highly depends on the available attribute predictors. By contrast, our method is more general and is able to find more diverse semantics. As shown in Fig.5(a), we successfully identify the directions corresponding to hair color, hair style, and skin color. This surpasses InterFaceGAN since predictors for these attributes are not easy to acquire in practice. Also, supervised methods are usually limited by the training objective. For example, InterFaceGAN is proposed to handle binary attributes. In comparison, our method can identify more complex attributes (*e.g.*, multiple eigen directions control the same attribute), like the different hair styles in Fig.5(b).



Figure 6: Real image manipulation with respect to various facial attributes. All semantics are found with the proposed *closed-form* approach, **SeFa**. GAN inversion [27] is used to project the target real image back to the latent space of StyleGAN [16].

Table 3: User study. We randomly generate 2,000 images for each dataset, and use the Top-50 eigen directions for each level of layers to manipulate these images. Users are asked how many directions result in *obvious* content change (numerator) and how many directions are semantically meaningful (denominator).

	Anime Face	Cat	Car	Church	Streetscape
Bottom Layers (0-1)	12/12	14/15	10/10	15/15	9/9
Middle Layers (2-5)	26/26	21/28	16/22	18/26	12/18
Top Layers (6-)	38/50	47/50	22/34	48/50	15/36

Real Image Manipulation. Following InterFaceGAN [23], we involve GAN inversion [9, 27] approaches into this work to achieve real image manipulation. More concretely, given a target image to edit, we first project it back to a latent code, and then use the discovered latent semantics to modulate the inverted code. Fig.6 shows some examples. We can tell that the semantics found by our *closed-form* approach are accurate enough to manipulate real images. For example, we manage to add eyeglasses to or remove eyeglasses from the input images (the fourth column of Fig.6).

3.3 Generalization to Other GAN Models

In this part, we verify the generalization ability of our algorithm by applying it to various types of state-of-the-art GAN models trained on diverse datasets.

Layer-wise Interpretation on StyleGAN and StyleGAN2. Our algorithm is general and flexible such that it can be used to interpret partial layers of the novel style-based generators [16, 17]. Hierarchical semantics identified from anime faces, cats, cars, churches, and streetscapes are shown in Fig.7. Recall that we do not have attribute predictors corresponding to these semantics. We also conduct a **user study** by manipulating images with the Top-50 eigen directions at each level and asking people how many directions lead to *obvious* content changes as well as how many directions are semantically meaningful. Tab.3 suggests that our *closed-form* solution can indeed find explanatory factors, even from some particular layers in GAN models.

Results on Conditional BigGAN. We also interpret the large-scale BigGAN [4] model, which is conditionally trained on 1,000 classes from ImageNet [6]. Note that BigGAN concatenates a category-derived embedding vector with the latent code to achieve conditional synthesis. Here, we only focus on the latent code part for semantic discovery. Fig.8 suggests that the semantics found by our *closed-form* algorithm can be faithfully applied to manipulating images from different categories. This verifies the generalization ability of our approach.

4 Conclusion

In this work, we propose a *closed-form* solution, **SeFa**, to factorizing the latent semantics learned by GANs. Extensive experiments demonstrate the effectiveness of our algorithm, which shows great power in identifying versatile semantics from different types of GAN models in an *unsupervised* manner.

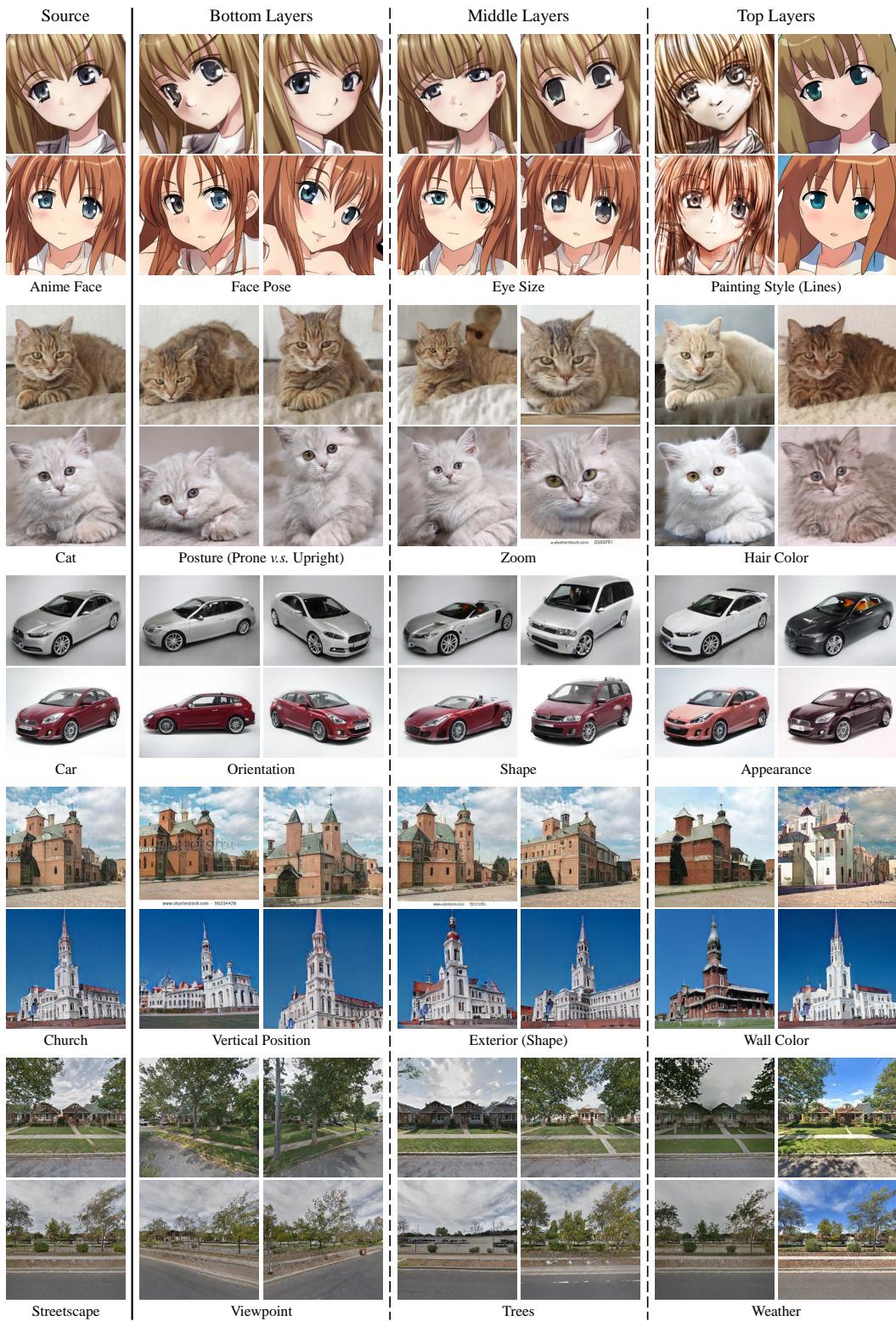


Figure 7: Hierarchical semantics discovered from style-based generators, *i.e.*, StyleGAN [16] and StyleGAN2 [17]. Among them, the streetscapes model is trained with StyleGAN2, while the others are using StyleGAN.

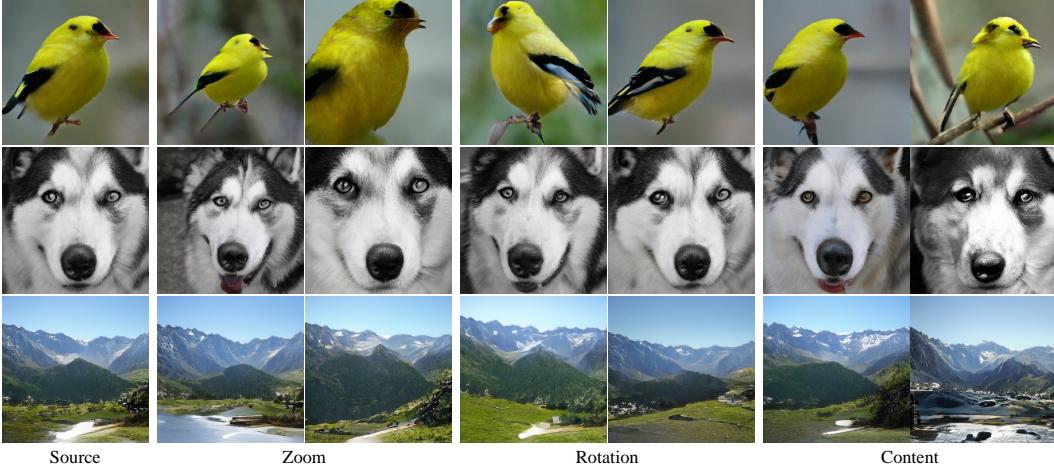


Figure 8: Diverse semantics found from BigGAN [4], which is conditionally trained on ImageNet [6]. These semantics are further used to manipulate images from different categories.

Broader Impact

This work would be of great interest to the general community from two aspects. First, it can show the great potential of reusing the knowledge learned by well-trained GAN models for a wide range of image editing applications without any retraining. Second, it conducts a pioneer study on the learned weight of GAN models, which will deepen our understanding on the internal mechanism of how GANs are able to produce photo-realistic images. This work follows the Explainable Artificial Intelligence (XAI), which aims at improving the interpretability of AI models and supporting the social right with the explanation for AI models' internal representations and decision making. On the other hand, however, it has the potential negative impact commonly existing in the topic of generative modeling for the misuse in the “deepfakes” relevant applications.

References

- [1] Anonymous, D. community, and G. Branwen. Danbooru2019: A large-scale crowdsourced and tagged anime illustration dataset. <https://www.gwern.net/Danbooru2019>, 2020.
- [2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.
- [3] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *ICLR*, 2019.
- [4] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.
- [5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [7] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019.
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.
- [9] J. Gu, Y. Shen, and B. Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020.

- [10] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [12] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.
- [13] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.
- [14] A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2020.
- [15] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.
- [16] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.
- [17] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019.
- [18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.
- [19] N. Naik, J. Philipoom, R. Raskar, and C. Hidalgo. Streetscore-predicting the perceived safety of one million streetscapes. In *CVPRW*, 2014.
- [20] J. Pamuła. Progressive training of gans with mutual information penalty. https://github.com/jonaszs/progressive_infogan, 2018.
- [21] A. Plumerault, H. L. Borgne, and C. Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020.
- [22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.
- [23] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.
- [24] A. Voynov and A. Babenko. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*, 2020.
- [25] C. Yang, Y. Shen, and B. Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *arXiv preprint arXiv:1911.09267*, 2019.
- [26] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [27] J. Zhu, Y. Shen, D. Zhao, and B. Zhou. In-domain gan inversion for real image editing. *ECCV*, 2020.