Real Image Editing on User-Specified Semantics by GAN Inversion model

Siting Li Si Jiang
IIIS, Tsinghua University
{li-st19, jiang-s18}@mails.tsinghua.edu.cn

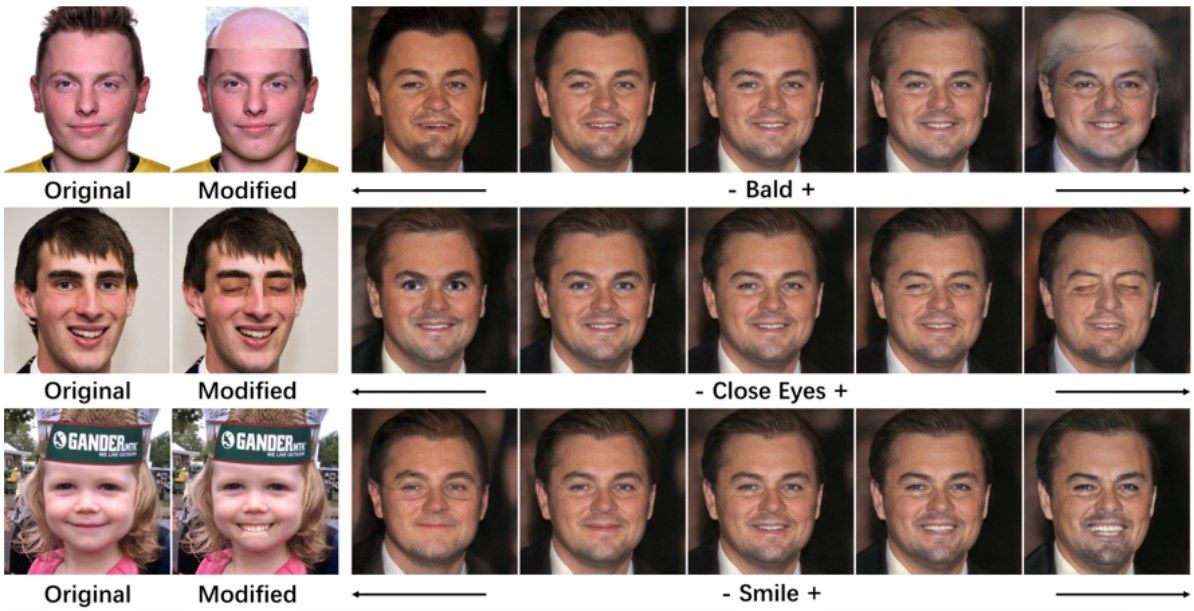

Figure 1. **Examples of various directions which are discovered by our framework.** For each row (each semantic), the left two images are the pairs used to find corresponding directions, right five images are the editing results on the real images by moving the latent code toward and backward the interpretable direction.

Abstract

Recent works have shown that there exists a variety of interpretable directions in the latent space of pre-trained Generative Adversarial Networks. In order to find such latent directions to control the semantics of synthesized images, previous works either adopt unsupervised methods, which are not strong enough to find disentangled directions, or supervised methods, which are not efficient and require a large number of labels. Also, most works focus on how to control the images directly synthesized by generators, little attention has been addressed on editing semantics on given real images. In this work, we proposed a compact framework to solve these two problems simultaneously. In particular, with the help of powerful GAN inversion models, we can obtain the latent directions for user-specified semantics with only one image pairs, and edit real images by directly manipulating latent codes. Extensive experiments show that directions found by our framework are comparable to the state-of-the-art unsupervised and supervised methods, and achieves satisfying real image editing. [1]

1. Introduction

Generative adversarial networks (GANs) [5] are widely used in various fields as a powerful generative model, such as image generation and manipulation [17] due to their ability to match the distribution of real-world datasets in an adversarial way. Variants of GAN, like PG-GAN [9], BigGAN [3], and StyleGAN [10], continue to improve the performance of the original GAN model, generating much

[1]Code and application are available at https://github.com/Su-Li-Fuwa/2021Spring-CV-Project

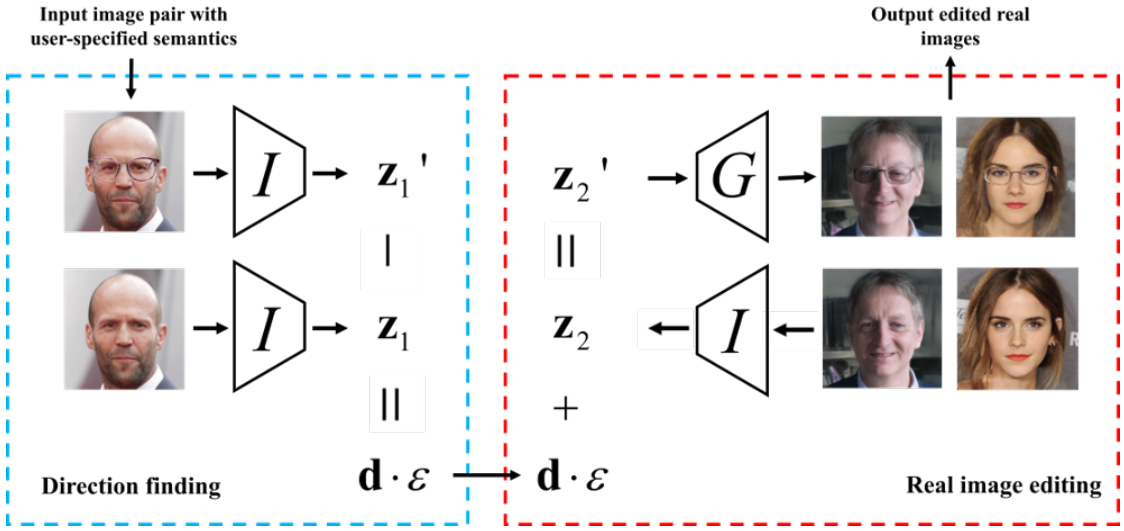

Figure 2. **Design architecture** of our framework. Blue box: direction finding for user-specified semantic. Red box: real image editing by manipulating on latent codes with given direction.

more realistic images from the given input latent vector. During training, GANs represent some attributes of images in their latent space. Recent researches show that there exist interpretable directions in pre-trained GANs' latent space, which can be used to control certain semantics of synthesized images, such as background color and object texture [4, 8, 20, 6, 19].

Efforts have been made to find interpretable directions in GANs' latent space. Some are supervised [4, 18], relying on random samplings and labels from statistics of images (e.g., color) or pre-trained attribute predictors, while others are unsupervised [19, 20, 6], based on mathematical assumptions of interpretable directions, such as orthogonality or being eigenvectors of the transformation matrix. However, due to the high-dimensionality of GANs' latent space and the variety of possible semantics, those works are not able to ensure the disentanglement of directions for different semantics, nor pick out directions for user-specified semantics, which may be useful in real-world applications.

Another attempt comes from the observation that most of the GANs are used to do image generation, what if we already have a bunch of real images, for example, photographs of human faces, can we do manipulation on these real images with help of GANs? This problem is closely related to the widely known *GAN inversion* problem, which reverses the generation process by mapping the image space back to the latent space [16, 1, 11, 22]. Recent work like In-Domain Inversion [22] takes both pixel level and semantic level information into account and reaches the state-of-the-art performance on GAN inversion. With perfect inverted latent codes, we can edit the real images by directly doing modifications in the latent space.

In this project, we propose a framework to do real image editing on user-specified semantics with the help of GAN inversion models. Concretely, by identifying a particular semantic (e.g., whether smiling or not) on only one image pair, our framework finds the corresponding direction in GANs' latent space. We study the latent space structure, finding that these directions are input-independent and can be generalized into all other real images, controlling the same semantic. We represent some interesting results in Fig.1. We mainly focus on StyleGAN [10], but our framework can be extended into variations of GANs such as PGGAN [9], BigGAN [3] by corresponding fine-tuned inversion models. We summarize our contributions as follows:

- We build a compact application to do real image editing, which takes advantage of pre-trained GANs and corresponding inversion models. The application only needs one pair of images to find the corresponding semantic and have amazing performance on image editing.
- We study the latent space structure with the help of state-of-the-art inversion models. By analyzing directions obtained from different real images with different semantics, we verify the input-invariance of these directions.

2. Related Works

Generative Adversarial Networks. Generative Adversarial Network (GAN) is a advanced and widely used generative model in image synthesis [5]. The generators in GANs take randomly sampled latent codes as the input and output high-fidelity images. The variations of GANs are trying to

control the properties of the synthesized images by adding a condition on latent code [14, 7, 24], and this idea is further improved by the style-based generator [10, 11]. However, interpretation of GANs' high-dimensional latent space and the mechanism of the generation process is much less explored.

Latent Semantic Interpretation. GANs are shown to represent semantics in the latent space [4, 8, 20, 6, 19]. Recent unsupervised methods include Harkonen *et al.* performing PCA on the sampled data to find primary directions in the latent space [6], and Shen *et al.* factoring on the first layer weight matrix and to find the most sensitive directions [19]. Both methods are efficient but perform poorly on finding disentangled directions for different semantics and can hardly do specified searching for user-specified semantics. Other methods that work in the supervised fashion can slightly alleviate this problem [18, 21, 4], but they heavily rely on the attribute predictors or human annotators to get labels. Voynov *et al.* proposed mathematical assumptions to ensure the disentanglement of directions [20], but at the same time hurt the performance on local semantics. Other works like StyleCLIP [15] can do searching for particular semantics, but it needs a huge pre-trained NLP model CLIP which can be costly and dataset-dependent.

GAN Inversion. GAN inversion [16, 1] is the core of real image editing, which finds the latent codes that most accurately recovering the images for a given GAN model. Previous works typically fall into two types. One is the learning-based method, which trains an inverse encoder under the supervision given by sampled codes and synthesized images [23, 16]. Another is the optimization-based method, which focuses on reconstructing one single image by directly minimize the pixel-wise loss [12, 13]. Recent works combine these two ideas by using the trained encoder to generate an initialization for single image optimization [2], and also consider the semantic level information during the inversion [22].

3. Method

3.1. Preliminaries

Mechanism of GANs. GANs are mainly composed of two parts: a generator $G(\cdot) : \mathcal{Z} \rightarrow \mathcal{X}$ which takes a latent code as the input and synthesizes a high-quality image; a discriminator $D(\cdot)$ which distinguishes real images and images synthesized by the generator. GANs are typically consisted of multiple layers with convolutional structures, transforming latent codes to high dimensional images step by step. Different from most GANs that directly input latent codes into the first layer of the generator, StyleGANs [10] first map the latent vector $z \in \mathcal{Z}$ to w with Multi-Layer Perceptron (MLP), which lies in a second latent space $\mathcal{W}$, and then map w into layer-wise style codes a with learned affine transformation.

Inversion of real images. Given a real image x^{real}, a GAN inversion model $I(\cdot) : \mathcal{X} \rightarrow \mathcal{Z}$, which tries to find the best latent code z^{inv} to recover x^{real}. State-of-the-art GAN inversion models first train an encoder that inverts x^{real} to a coarse latent code z^{init}, then fine-tune on z^{init} by reducing the pixel-wise difference to obtain accurate z^{inv}. Inspired by the phenomenon that the latent space of GANs encodes rich semantic knowledge, In-domain GAN inversion [22] takes the semantic level loss into account in both encoder training and latent code fine-tuning phases and makes z^inv align with the semantic space S learned in the pre-trained GAN model.

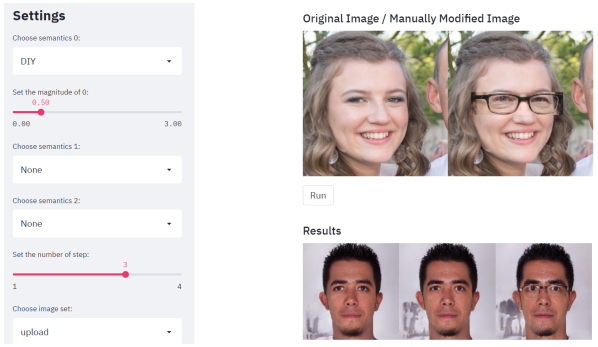

Figure 3. **Interface** for real image editing.

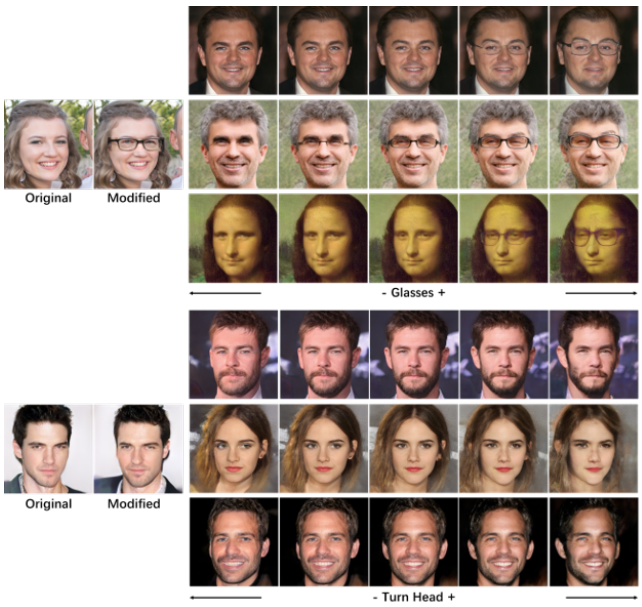

Figure 4. Examples of various semantic directions discovered by our framework.

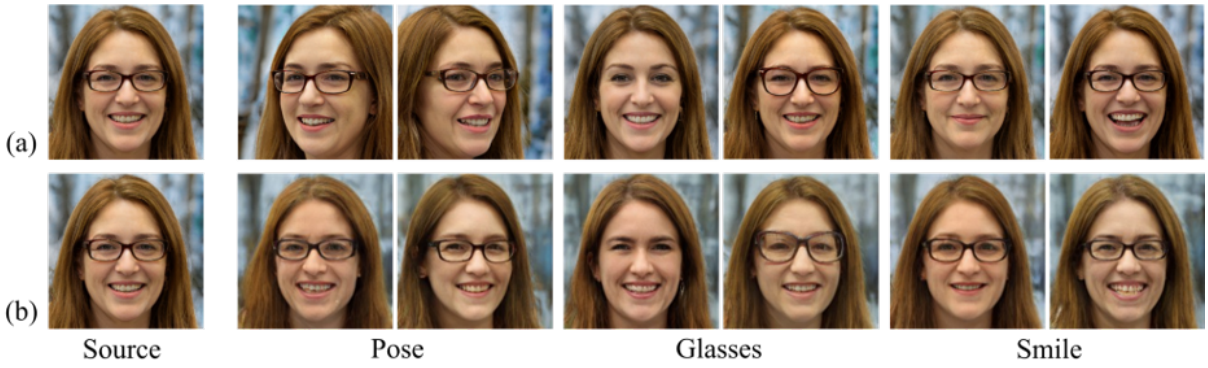

Figure 5. Qualitative comparison between (a) InterFaceGAN [18] and (b) ours on FFHQ dataset.

Figure 6. Qualitative comparison between (a) GANSpace [6] and (b) ours on FFHQ dataset.

3.2. Implementation

Our framework, which is shown in 2, is mainly composed of two parts. The first part is direction-finding, which finds the direction in the latent space for a user-specified semantic by the following approach: Given a real image x_1, the user manually modifies it and obtains a new image x_1', which is different from x_1 by a particular semantic (for example, on FFHQ dataset [10], x_1' can be obtained from adding glasses or erasing earrings on x_1). This image pair $\{x_1, x_1'\}$ will input into In-domain inversion model I to get latent codes $\{z_1, z_1'\}$ and the corresponding direction of this specified semantic is obtained by $d = \frac{z_1' - z_1}{||z_1' - z_1||}$. The second part is real image editing by directly manipulating latent codes with given direction: given an arbitrary real image x_2, we first invert it into the latent code $z_2 = I(x_2)$. After applying the direction d on z_2, we can synthesize a new image $\text{edit}(x_2) = x_2' = G(z_2 + d \cdot \epsilon)$ with modified semantic. ϵ denotes the manipulation intensity and can be either positive or negative. For StyleGAN models, we adjust the framework a little bit by inverting the real image into second latent space $\mathcal{W}$ instead of $\mathcal{Z}$. For convenience, we develop an interface to enable human-model interaction, which is shown in Fig.3. With this interface, users can edit given or self-modified semantics on default or self-uploaded real images.

We would like to address attention on the point that our application only needs one image pair to find the corresponding direction, and this direction can be performed on all real images to obtain the same effect on semantic modification. This indicates that the direction found by one particular point in $\mathcal{Z}$ can be generalized to all other points in $\mathcal{Z}$, which indicates the input-invariant property. Also, by performing the linear combination of two directions on the latent code, we find both semantics have been modified on the real image, which indicates the linear superposition property. We study these interesting properties, which imply the structure of GAN's latent space, and some results can be found in section 4.4.

4. Experiments

We evaluate our framework to discover the interpretable directions in GAN's latent space on a wide range of semantics. We also compare our method with existing supervised and unsupervised alternatives and demonstrate its effectiveness on real image editing. In addition, we use the framework to discover how different semantics are encoded in GAN's latent space and describe some interesting discoveries from our analysis.

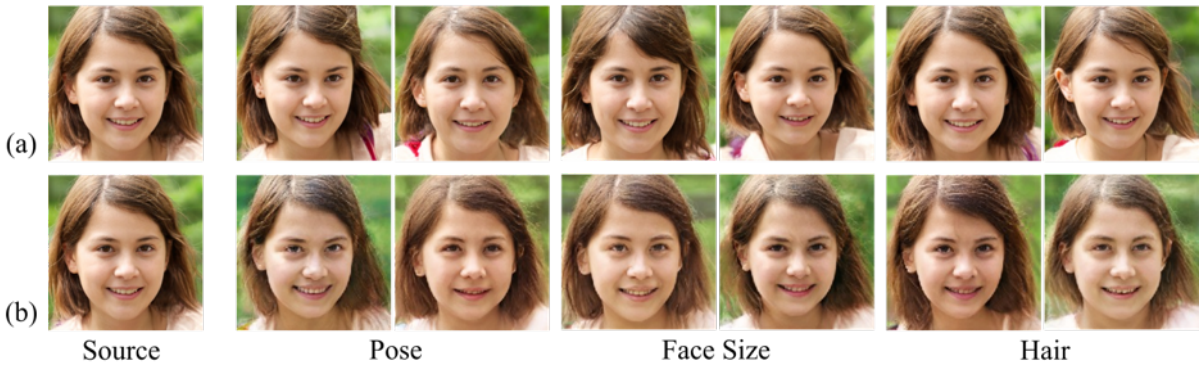

Figure 7. Qualitative comparison between (a) SeFa [19] and (b) ours on FFHQ dataset.

4.1. Results on StyleGAN2 and FFHQ Dataset

We evaluate our framework on a StyleGAN2 model pre-trained on the FFHQ dataset [10], containing 70,000 high-quality images at 1024 resolution and a lot of variety in terms of age, ethnicity, and viewpoint. We focus on the semantics of the human face because of the good definition of facial attributes. Fig.4 shows the semantic directions found by our framework. Since the semantics are set by users, there are plenty of available semantic directions due to the various input. We find that the direction calculated on a single instance can be applied to manipulating different images with good performance, which implies the invariant property of semantic directions. Further, because our framework requires very weak supervision (only one input instance for defining the semantics), it is time-saving and resource-saving.

4.2. Comparison with Previous Approaches

We compare our framework to state-of-the-art supervised and unsupervised approaches, including InterFaceGAN [18], GANSpace [6], and SeFa [19] on StyleGAN2 model pre-trained on FFHQ dataset. The major difference is that our framework can deal with user-specified semantics, while others either are confined to pre-defined semantics or random and thus uncontrollable semantics.

Comparison with supervised approaches. We choose InterFaceGAN [18] for comparison. Fig.5 visualizes some qualitative comparison results where we observe that our framework successfully finds the semantic direction found by InterFaceGAN and its performance is quite good, suggesting that weak supervision can achieve similar performance to the supervised model in this task. Since GAN inversion can be done more efficiently than sampling plenty of data and pre-training attribute predictors, our framework is more suitable for an application, where various semantic directions are required in real time. However, since there is weak supervision in our framework when facing large changes like turning head, it is hard for our framework to capture the semantics and find corresponding directions. In the instance we show, it can only change the face orientation a little.

Comparison with unsupervised approaches. We choose GANSpace [6] and SeFa [19] for comparison. GANSpace performs PCA on sampled data to find principle directions in the latent space, while SeFa performs eigenvalue decomposition on the first projection done by GAN models to find directions that cause large variations after projection. Both methods propose some assumptions about the interpretability of directions. Fig.6 and Fig.7 visualize some qualitative comparison results. We observe that our framework finds more disentangled directions: When changing the face pose, GANSpace changes the background as well, while we preserve the background; when changing the face size, SeFa changes the hair length at the same time. More importantly, GANSpace and SeFa find directions for random semantics, and sometimes the semantics are highly entangled or artifact-causing and thus hard to interpret. This implies that the assumptions they propose may not hold for interpretable directions. For example, directions that cause large variations may signify artifacts and produce outputs of low quality.

4.3. Failure Analysis

To find the limitation of our framework, we analyze the failure cases in experiments and divide the failures into four cases, as shown in Fig.8.

- Case 1 happens when the input semantics is unusual in the training dataset. For example, when the semantics is “Close One Eye”, it fails to find the direction, since it is hard to do inversion for unusual images and possibly there is no direction for unusual semantics in the latent space.

- Case 2 takes place when the input pair defines complex semantics, which is a combination of multiple basic semantics. In the instance we provided, the change

includes skin color, gender, hair, face size, and facial expression. When we apply the direction to other images, heavy artifacts appear and influence the image quality.

- Case 3 shows a common problem in finding disentangled directions in latent representation. For instance, we would like to find the semantic direction for "Shave Beard", but the found direction accidentally changes the smile and the glasses when "shaving beard", resulting in mixed semantics. In section 4.4, we propose an effective and efficient solution for solving this problem.

- Case 4 shows another problem in applying directions. Some semantics cannot be added to certain faces, or it will lead to artifacts or other unwanted changes. For example, "Beard" can be added to male faces but not female faces.

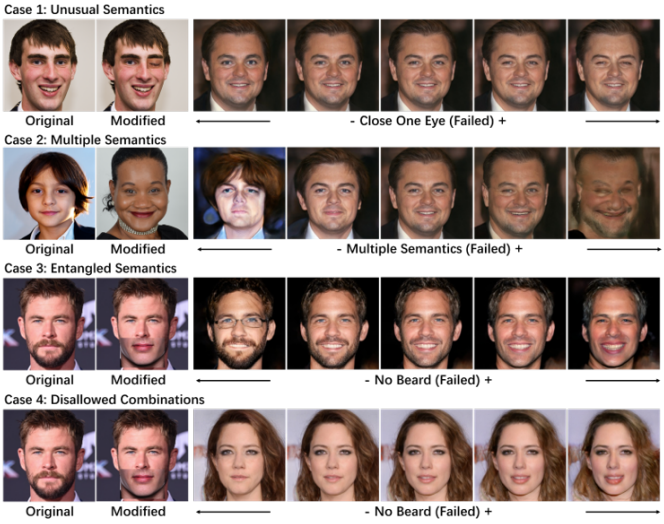

Figure 8. Four cases of failure of our framework.

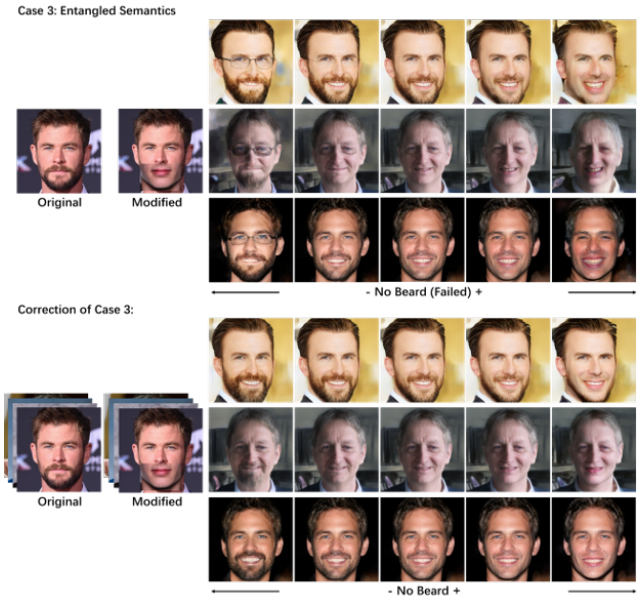

Figure 9. Comparison of direction found by one pair of input and direction found by 10 pairs of input.

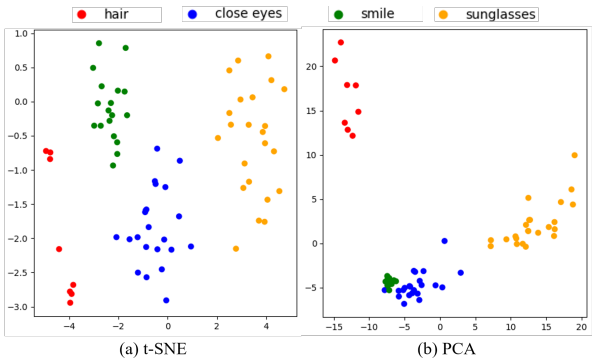

Figure 10. Results of t-SNE and PCA embedding of latent vectors. The t-SNE embeddings and the PCA embeddings do not use the class labels, which are only used during visualization.

4.4. Discover StyleGAN's Latent Space

Motivated by the invariant property of directions we observe in practice and the convenience of our framework, we conduct experiments to discover the structure of StyleGAN's latent space. We collect a bunch of input pairs, calculate the direction they define, and perform two-dimensional t-SNE and PCA embedding of these directions. The results are visualized in Fig.10. We find that the directions that induce the same semantics cluster in the latent space. Since the input pairs vary in gender, skin color, and so on, this implies that the direction for a certain semantics has some kind of invariant property and is independent of inputs to some degree.

Can we use the invariant property to improve the quality of directions, such as realize more disentangled semantic change? We further calculate the average of each cluster and apply the average direction to real images. Fig.9 shows the improvement brought by this strategy. The average direction yields a more disentangled effect than the original effect: The glasses and smile attributes remain unchanged along the average direction, while the beard is still "shaved". This high-quality direction is found by adding more samples for supervision (in this experiment, 10 samples are used), but the supervision is far weaker than the previous supervised method, showing the advantage of our method over alternatives.

5. Conclusions and Discussion

We propose a framework to do real image editing on user-specified semantics by GAN inversion. Our results show that it finds user-defined semantic direction efficiently and with high accuracy, compared with previous approaches. Our further experiments reveal the input-independence of semantic directions, which allows us to apply directions to various real images and achieve satisfying results. Our study on the generator's latent space is straightforward, which only verifies some basic properties of the latent space. There are lots of interesting works that can be further conducted. For example, by the properties of linear additivity, we can hopefully decompose one entangled direction, which changes multiple semantics, into some predetermined disentangled directions for common semantics. Besides, knowing the structure of pre-trained GANs' latent space can help us to design better regularizers and enhance the disentanglement during the training process.

References

[1] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics*, 38(4):1–11, Jul 2019. 2, 3

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate, 2019. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019. 1, 2

[4] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties, 2019. 2, 3

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 1, 2

[6] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls, 2020. 2, 3, 4, 5

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 3

[8] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks, 2020. 2, 3

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. 1, 2

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 1, 2, 3, 4, 5

[11] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. 2, 3

[12] Zachary C. Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks, 2017. 3

[13] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. 2019. 3

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 3

[15] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021. 3

[16] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing, 2016. 2, 3

[17] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1

[18] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 2, 3, 4, 5

[19] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans, 2020. 2, 3, 5

[20] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space, 2020. 2, 3

[21] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis, 2020. 3

[22] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing, 2020. 2, 3

[23] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold, 2018. 3

[24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. 3