Rethinking the editing of generative adversary networks: a method to estimate editing vectors based on dimension reduction

Haoran Jiang
CS undergraduate
SIST ShanghaiTech University
jianghr1@shanghaitech.edu.cn

Zhenghong Yu*
CS undergraduate
SIST ShanghaiTech University
yuzhh1@shanghaitech.edu.cn

Abstract

While Generative Adversarial Networks (GANs) have recently found applications in image editing, most previous GAN-based image editing methods require large-scale datasets with semantic segmentation annotations for training, only provide high-level control, or merely interpolate between different images. Previous researchers have proposed EditGAN [30]for high-quality, high-precision semantic image editing with limited semantic annotations by finding 'editing vectors'. However, it is noticed that EditGAN [30]consumes a lot of hardware resources to train and optimize. Based on the orthogonality of latent space observed by EditGAN [30], we propose a method to estimate editing vectors that do not rely on semantic segmentation nor differentiable feature estimation network. Our method assumes that there is a correlation between the intensity distribution of features and the distribution of hidden vectors, and estimates the relationship between the above distributions by sampling the feature intensity of the image corresponding to several hidden vectors. We then found that this method has a good effect in processing different kinds of features.

1 Introduction

Image Editing and Manipulation. GAN-based image editing methods can be broadly sorted into a number of categories. (i) One line of work relies on the careful dissection of the GAN's latent space, aiming to find interpretable and disentangled latent variables, which can be leveraged for image editing, in a fully unsupervised manner [2, 5, 6, 18, 19, 23, 37, 42–46]. Although powerful, these approaches usually do not result in any high-precision editing capabilities. The editing vectors we are learning in EditGAN [30]would be too hard to find independently without segmentation-based guidance. (ii) Other works utilize GANs that condition on a class or pixel-wise semantic segmentation labels to control synthesis and achieve editing [9, 11, 26, 34, 47, 48, 56]. Hence, these works usually rely on large annotated datasets, which are often not available, and even if available, the possible editing operations are tied to whatever labels are available. This stands in stark contrast to EditGAN [30], which can be trained in a semi-supervised fashion with very little labeled data and where an arbitrary number of high-precision edits can be learned. (iii) Furthermore, auxiliary attribute classifiers have been used for image manipulation [20, 21], thereby still relying on annotated data and usually only providing high-level control. (iv) Image editing is often explored in the context of "interpolating" between a target and different reference images in sophisticated ways, for example by replacing certain features in a given image with features from a reference image [12, 25, 27, 56]. From the general image editing perspective, the requirement of reference images limits the broad

*Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

applicability of these techniques and prevents the user from performing specific, detailed edits for which potentially no reference images are available. (v) Recently, different works proposed to directly operate in the parameter space of the GAN instead of the latent space to realize different edits [4, 5, 10]. For example, [4, 5] essentially specialize the generator network for certain images at test time to aid image embedding or "rewrite" the network to achieve desired semantic changes in output. The drawback is that such specializations prevent the model from being used in real-time on different images and with different edits. [10] proposed an approach that more directly analyses the parameter space of a GAN and treats it as a latent space in which to apply edits. However, the method still merely discovers edits in the network's parameter space, rather than actively defining them as we do. It remains unclear whether their method can combine multiple such edits, as we can, considering that they change the GAN parameters themselves. (vi) Finally, another line of research targets primarily very high-level image and photo stylization and global appearance modifications [17, 24, 29, 32–35, 47, 51].

Generally, most works only do relatively high-level and not detailed, high-precision editing, which our model targets. Hence, we consider our model complementary to this body of work.

GANs and Latent Space Image Embedding. EditGAN [30]builds on top of DatasetGAN [53] and SemanticGAN [28], which proposed to jointly model images and their semantic segmentation using shared latent codes. However, these works leveraged this model design only for semi-supervised learning, not for editing. EditGAN [30]also relies on an encoder, together with optimization, to embed new images to be edited into the GAN's latent space. This task in itself has been studied extensively in different contexts before, and we are building on these works. Previous papers studied encoder-based methods [8, 15, 16, 36, 40], used primarily optimization-based techniques [1, 14, 22, 31, 37, 39, 50, 55], and developed hybrid approaches [5–7, 54, 55].

Finally, a concurrent paper [49] shares similarities with DatasetGAN [53], on which our method builds, and explores an editing approach related to EditGAN [30]as one of its applications. However, our editing approach is methodologically different and leverages editing vectors, and also demonstrates significantly more diverse and stronger experimental results. Furthermore, [3] shares some high level ideas with EditGAN [30]; however, it leverages the CLIP [38] model and targets text-driven editing.

2 Existing works

2.1 GAN and latent space image embedding

DatasetGAN [53] and SemanticGAN [28] proposed to jointly model images and their semantic segmentations using shared latent codes. For the generative advisory model,

$$\text{Generator: } \mathcal{W}^+ \to \mathcal{X}$$

$$\text{Discriminator: } \mathcal{X} \to \mathrm{score}[0,1]$$

The generator generates pictures of image space $\mathcal{X}$ from latent space $\mathcal{W}^+$, while the discriminator classified the generated pictures to different label. However, these works leveraged this model design only for semi-supervised learning, not for editing.

2.2 EditGAN

EditGAN [30]lies in leveraging the joint distribution of images and semantic segmentations for high-precision image editing. EtitGAN tries to embed new images into latent space and generate the corresponding segmentation output. The new model is,

$$\text{Generator: } \mathcal{W}^+ \to \mathcal{X}, \mathcal{Y}$$

$$\text{Discriminator: } \mathcal{X}, \mathcal{Y} \to \mathrm{score}[0,1]$$

while $\mathcal{Y}$ represents semantic space. EditGAN [30]denotes the edited segmentation mask by $\mathbf{y}_{\text{edited}}$. Starting from the embedding $\mathbf{w}^+$ of the unedited image $\mathbf{x}$ and segmentation $\mathbf{y}$, EditGAN [30]then perform optimization within $\mathcal{W}^+$ to find a new $\mathbf{w}^+_{\text{edited}} = \mathbf{w}^+ + \delta\mathbf{w}^+_{\text{edit}}$ consistent with the new segmentation $\mathbf{y}_{\text{edited}}$, while allowing the RGB output $\mathbf{x}$ to change within the editing region.

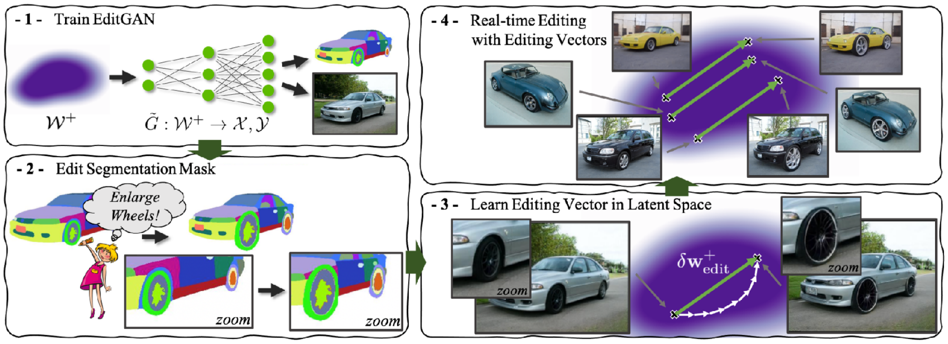

Figure 1: Pipeline of EditGAN

EditGAN [30]formally seeks an *editing vector* $\delta\mathbf{w}^+_{edit} \in \mathcal{W}^+$ such that $(\mathbf{x}_{edited}, \mathbf{y}_{edited} = \tilde{G}(\mathbf{w}^+ + \delta\mathbf{w}^+_{edit})$, where $\tilde{G}$ denotes the fixed generator that synthesizes both images and segmentations. To find $\delta\mathbf{w}^+$, approximating $\delta\mathbf{w}^+_{edit}$, EditGAN [30]use the following losses as minimization targets:

$$\begin{aligned}\mathcal{L}_{\text{RGB}}(\delta\mathbf{w}^+) = L_{\text{LPIPS}}(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta\mathbf{w}^+) \odot (1-r), \mathbf{x} \odot (1-r)) \\ + L_{L2}(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta\mathbf{w}^+) \odot (1-r), \mathbf{x} \odot (1-r))\end{aligned} \tag{1}$$

$$\mathcal{L}_{\text{CE}}(\delta\mathbf{w}^+) = H(\tilde{G}^{\mathbf{y}}(\mathbf{w}^+ + \delta\mathbf{w}^+) \odot r, \mathbf{y}_{\text{edited}} \odot r) \tag{2}$$

$$\mathcal{L}_{\text{ID}}(\delta\mathbf{w}^+) = \langle R(\tilde{G}^{\mathbf{x}}(\mathbf{w}^+ + \delta\mathbf{w}^+), R(\mathbf{x}))\rangle \tag{3}$$

where H denotes the pixel-wise cross-entropy, L_{LPIPS} loss is base on the Learned Perceptron Image Patch Similarity (LPIPS) distance [52], and L_{L2} is a regular pixel-wise L2 loss. $\mathcal{L}_{\text{RGB}}(\delta\mathbf{w}^+)$ ensures that the image appearance does not change *outside* the region of interest, while $\mathcal{L}_{\text{CE}}(\delta\mathbf{w}^+)$ ensures that the target segmentation $\mathbf{y}_{\text{edit}}$ is enforced *within* the editing region. When editing human fances, EditGAN [30]also apply the identity loss $\mathcal{L}_{\text{ID}}(\delta\mathbf{w}^+)$, with R denoting the pretrained ArcFace feature extraction network and $\langle\cdot,\cdot\rangle$ cosine-similarity.

The final objective function for optimization then becomes:

$$\mathcal{L}_{\text{editing}}(\delta\mathbf{w}^+) = \lambda_1^{\text{editing}}\mathcal{L}_{\text{RGB}}(\delta\mathbf{w}^+) + \lambda_2^{\text{editing}}\mathcal{L}_{\text{CE}}(\delta\mathbf{w}^+) + \lambda_3^{\text{editing}}\mathcal{L}_{\text{ID}}(\delta\mathbf{w}^+) \tag{4}$$

with hyperparameters $\lambda_{1,\cdots,3}^{\text{editing}}$. The only "learnable" variable is the editing vector $\delta\mathbf{w}^+$; all neural networks are kept fix. After optimizing $\delta\mathbf{w}^+$ with the objective function, we can use $\delta\mathbf{w}^+ \approx \delta\mathbf{w}^+_{edit}$. EditGAN [30]rely on the generator, trained to synthesize realistic images, to modify the RGB values in the editing region in a plausible way consistent with the segmentation edit.

Summarizing, EditGAN [30]perform image editing with EditGAN [30]in three different modes:

- **Real-time Editing with Editing Vectors.** For localized, well-disentangled edits EditGAN [30]perform editing purely by appling previously learnt editing vectors with varying scales s_{edit} and manipulate images at interactive rates.
- **Vector-based Editing with Self-Superised Refinement.** For localized edits that are not perfectly disentangled with other parts of the image, we can remove editing artifacts by additional optimization at test time, while initializing the edit using the learnt editing vectors.
- **Optimization-based Editing.** Image-specific and very large edits do not transfer to other images via editing vectors. For such operations, EditGAN [30]perform optimization from sratch.

2.3 Shortage of EditGAN

Although EditGAN [30]pays attention to the orthogonality of latent space, it only uses this property to propose the method of editing vectors, and does not make use of the properties of space in the process of solving editing vectors.

Besides, as we have introduced above, EditGAN [30]is an optimization based approach which requires plenty handful annotations and heavy training, especially when training its semantic branch.

3 Methods

3.1 Latent space analysis

InterfaceGAN [43] proposed that hyperplane can be found in latent space to classify whether the image generated by any latent vector has specific features, which inspired us to consider the relationship between the hyperplane found in latent space to classify features and the corresponding editing vector.

One of the major differences between our case and InterfaceGAN [43] is the optimization problem. InterfaceGAN [43] uses Support Vector Machines (SVM) [13] to classify existence of feature in latent space, which is to maximize interval between two classes. In our case, the objective function is slightly different, which is to find the vector from the center of one class to another.

Another major difference is the number of annotations. InterfaceGAN [43] originally uses 6000 fine-annotated samples (12 times of the dimension of latent space), while in we uses only 1000 samples (2 times of the dimension of latent space). Which lead us to dimension reduction methods rather than learning base methods.

3.2 Dimension reduction in latent space

The features we deal with fall into two categories: binary features which has only two states (have or not have); continuous features which can be regarded to has strength from 0 to 1.

3.2.1 Binary feature editing

We introduced Linear Discrimination Analysis(LDA) [41] to handle binary feature. In the binary classification problem, LDA [41] can robustly estimate the projection vector that maximizes the distance between classes and minimizes the variance within a class after data points are projected. For binary features, direction of editing vector can directly obtain by performing binary class LDA [41]:

Let D_0, D_1 denote latent vectors that are in class 0 and class1 (i.e. not has or has certain feature), let $\boldsymbol{\mu}_0$, $\boldsymbol{\mu}_1$ denote the mean of latent vectors in D_0, D_1 respectively. Let $\mathbf{S}_b$, $\mathbf{S}_w$ denote the between class and within class scatter matrix which is defined as:

$$\mathbf{S}_b = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T$$

$$\mathbf{S}_w = \sum_{k\in\{0,1\}} \left(\sum_{\mathbf{w}_i\in D_k} \mathbf{w}_i - \boldsymbol{\mu}_k\right)\left(\sum_{\mathbf{w}_i\in D_k} \mathbf{w}_i - \boldsymbol{\mu}_k\right)^T$$

And the direction of editing vector $\delta\mathbf{w}$ can be obtain by solving the convex optimization problem below:

$$\begin{aligned} \arg\min_{\delta\mathbf{w}} \quad & -\delta\mathbf{w}^T\mathbf{S}_b\delta\mathbf{w} \\ \text{s.t.} \quad & \delta\mathbf{w}^T\mathbf{S}_w\delta\mathbf{w} = 1 \end{aligned} \tag{5}$$

It is easy to proof that the optimal $\delta\mathbf{w}$ is equal to the eigenvector related to the maximum eigenvalue of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The effectiveness of LDA [41] can be easily verified by precision on validation sets, and the editing vector can than be obtained by:

$$\delta\mathbf{w}^+_{\text{edit}} = \delta\mathbf{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\delta\mathbf{w} \tag{6}$$

3.2.2 Continuous feature editing

Estimating editing vector for continuous features is far more tricky. Here we proposed two approaches that are bipolar method and discretizing method.

Bipolar method is to split the dataset into low, medium and high strength parts, and let the low strength part be class 0 and high strength part be class 1 to obtain estimation of editing vector. This method is quite straightforwards but not accurate nor efficient.

Discretizing method also split the dataset into parts but with more bins. Ordinarily we use a setting with 5 groups: $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$ and $[0.8, 1]$. Rather than performing multi-class LDA [41] to obtain a class by class discriminator, we set the optimization problem to find a single projection vector that maximize between class scatter and minimize within class scatter when projected on it. The normal form of the optimization problem is define as below:

Let D_k denotes the set of samples that are in the k-th class, $\boldsymbol{\mu}_k$ denotes the mean vector of the samples in the k-th class, $\overline{\boldsymbol{\mu}}$ denotes the mean vector of all samples. Follow the definition in binary cases, let $\mathbf{S}_b$, $\mathbf{S}_w$ denote the between class and within class scatter matrix which is defined as

$$\mathbf{S}_b = \sum_k (\boldsymbol{\mu}_k - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_k - \overline{\boldsymbol{\mu}})^T$$

$$\mathbf{S}_w = \sum_k \sum_{\mathbf{w} \in D_k} (\mathbf{w} - \boldsymbol{\mu}_k)(\mathbf{w} - \boldsymbol{\mu}_k)^T$$

The convex optimization problem is in the form of:

$$\begin{aligned} \arg\min_{\delta\mathbf{w}} \quad & -\delta\mathbf{w}^T \mathbf{S}_b \delta\mathbf{w} \\ \text{s.t.} \quad & \delta\mathbf{w}^T \mathbf{S}_w \delta\mathbf{w} = 1 \end{aligned} \tag{7}$$

Similarly we have the optimal $\delta\mathbf{w}$ be the eigenvector related to the maximum eigenvalue of $\mathbf{S}_w^{-1}\mathbf{S}_b$, and it is necessary to verify if the projection length of latent vector is in direct proportion to the strength of feature. The editing vector can be obtained by:

$$\delta\mathbf{w}^+_{\text{edit}} = \frac{\delta\mathbf{w}^T(\boldsymbol{\mu}_4 - \boldsymbol{\mu}_0)\delta\mathbf{w}}{0.8} \tag{8}$$

4 Experiments

As GAN based editing methods require a well-trained GAN network as base model, we choose StyleGAN-Human as our base GAN model.

4.1 EditGAN methods

According to the method of EditGAN [30], we manually labeled 100 groups of human body segmentation data, trained semantic branches, and then obtained the editing vectors of the upper garment lengths. The editing results by vectors obtained by EditGAN [30]methods are shown in Fig 2.

Figure 2: Editing results of EditGAN [30]methods.

Figure 3: Editing results of binary feature

Figure 4: Editing results of continuous feature

Metrics	EditGAN[30]	Ours
Correlation	0.5788	0.3911
L2 distance	1.1214	

Table 1: Cross validation results between EditGAN and ours

4.2 Proposed methods

Binary feature editing We set whether there is pattern on cloth as the target feature and labeled 500 images. According to the method discussed above, we perform LDA [41] over the sample sets and obtain the editing vector. The editing results are shown in Fig 3.

Continuous feature editing We use the length of upper garment as target feature to compare with EditGAN [30]method and labeled 500 images. Then the discretizing method was performed according to the method discussed above to obtain the editing vector. The editing results are shown in Fig 4.

4.3 Cross validation

As both methods estimate the editing vector for length of upper garment, it is possible to compare their effectiveness on feature strength estimation and consistency. We compared the correlation between projection length and feature strength and the L2 distance between two editing vector. The results are shown in Table 1.

5 Conclusion

We have developed an efficient GAN image editing technology based on the vector space editing technology proposed by EditGAN [30], and proved that our method can efficiently find the general direction of the editing vector under its limited annotation data. Compared with EditGAN [30], our method explicitly has the ability to edit discrete binomial features; Compared with InterfaceGAN [43], our method proposes an efficient method to find edit vectors on continuous features. At the same time, we noticed that the editing vector we found was not completely decoupled from other features, and some other features also changed in the editing process. In the future, we plan to explore more explicit editing vector estimation methods on continuous features. We also plan to better solve the decoupling problem of editing vectors in finite annotation.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5134–5142, 2020.

[3] David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. Paint by word. *arXiv preprint arXiv:2103.10951*, 2021.

[4] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *European conference on computer vision*, pages 351–369. Springer, 2020.

[5] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

[7] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019.

[8] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[9] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: Deep generation of face images from sketches. *ACM Transactions on Graphics (TOG)*, 39(4):72–1, 2020.

[10] 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.

[11] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[12] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020.

[13] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[14] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

[15] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[16] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[18] 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.

[19] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020.

[20] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

[21] Xianxu Hou, Xiaokang Zhang, Hanbang Liang, Linlin Shen, Zhihui Lai, and Jun Wan. Guidedstyle: Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks*, 145:209–220, 2022.

[22] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *European Conference on Computer Vision*, pages 17–34. Springer, 2020.

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

[24] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 848–856. IEEE, 2019.

[25] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 852–861, 2021.

[26] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.

[27] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Vogue: Try-on by stylegan interpolation optimization. 2021.

[28] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8300–8311, 2021.

[29] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018.

[30] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In *Advances in Neural Information Processing Systems (NeurIPS)*,

2021.
[31] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.
[32] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017.
[33] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4990–4998, 2017.
[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.
[35] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.
[36] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.
[37] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *arXiv preprint arXiv:2001.10238*, 2020.
[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
[39] Ankit Raj, Yuqi Li, and Yoram Bresler. Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5602–5611, 2019.
[40] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.
[41] Robert Harry Riffenburgh. *Linear discriminant analysis*. PhD thesis, Virginia Polytechnic Institute, 1957.
[42] 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.
[43] 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.
[44] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021.
[45] 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.
[46] Binxu Wang and Carlos R Ponce. The geometry of deep generative image models and its applications. *arXiv preprint arXiv:2101.06006*, 2021.
[47] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.
[48] Rongliang Wu, Gongjie Zhang, Shijian Lu, and Tao Chen. Cascade ef-gan: Progressive facial expression editing with local focuses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5021–5030, 2020.
[49] Jianjin Xu and Changxi Zheng. Linear semantics in generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9351–9360, 2021.
[50] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.
[51] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9036–9045, 2019.
[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
[53] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.
[54] 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.
[55] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pages 597–613. Springer, 2016.
[56] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.