Self-Conditioned Generative Adversarial Networks for Image Editing

Yunzhe Liu\textsuperscript{1} Rinon Gal\textsuperscript{2} Amit H. Bermano\textsuperscript{2} Baoquan Chen\textsuperscript{1} Daniel Cohen-Or\textsuperscript{2}

CFCS, Peking University\textsuperscript{1} Tel Aviv University\textsuperscript{2}

Abstract

Generative Adversarial Networks (GANs) are susceptible to bias, learned from either the unbalanced data, or through mode collapse. The networks focus on the core of the data distribution, leaving the tails — or the edges of the distribution — behind. We argue that this bias is responsible not only for fairness concerns, but that it plays a key role in the collapse of latent-traversal editing methods when deviating away from the distribution’s core. Building on this observation, we outline a method for mitigating generative bias through a self-conditioning process, where distances in the latent-space of a pre-trained generator are used to provide initial labels for the data. By fine-tuning the generator on a re-sampled distribution drawn from these self-labeled data, we force the generator to better contend with rare semantic attributes and enable more realistic generation of these properties. We compare our models to a wide range of latent editing methods, and show that by alleviating the bias they achieve finer semantic control and better identity preservation through a wider range of transformations. Our code and models will be available at https://github.com/yzliu567/sc-gan.

1. Introduction

Generative Adversarial Networks [14] (GANs) have shown remarkable performance on a wide range of synthesis-related tasks. Recently, the structure of their latent space has been thoroughly explored, giving birth to a wide range of methods designed to manipulate the generated images in semantically meaningful ways. However, GANs tend to suffer from two kinds of biases. The first, and most intuitive type of bias, is the biased learned from the data. If two attributes are highly correlated in the data (e.g., glasses and age) the network learns to tie them together, leading to entanglement in the latent space. The second kind of bias is an inherent Generative Bias, so called because it is derived from the training data itself. We argue that this Generative Bias plays a key role in the poor performance of GANs when tasked with synthesizing such rare instances. We focus on latent-based image manipulations, and postulate that the failures of classical GAN-based editing techniques can be traced in part to this same root. We propose to tackle this bias by converting a pre-trained generator into a self-conditioned model [21], where continuous conditioning labels are derived from the latent space of the generator itself. In doing so, we build on existing linear editing methods and achieve superior quality, control and identity preservation.

Our method consists of four steps: First, we find a linear editing direction responsible for a faulty attribute which we wish to enhance. Such directions can be found with weak supervision [32], in a zero-shot manner [24] or even in an un-
supervised fashion [16, 33]. Second, we build on prior observations that latent space distances are linearly correlated with semantic attribute strengths [22]. We can thus label the original training set by inverting all images into the latent space of the network and calculating their latent projections on the editing direction. Third, we convert the generator and the discriminator into conditional variants, where we condition the generation on the latent-space distance labels. Finally, we fine-tune the network with samples that are drawn uniformly according to their latent-distance labels.

By following this approach, the conditional model is penalized for ignoring the edges of the distribution, drawing it towards a more uniform output distribution. Moreover, the editing directions are baked into the model’s architecture, allowing for better semantic control. Our method additionally enables multi-attribute editing, even for those cases where latent traversal often fails. We validate our approach through a set of experiments and demonstrate that it leads to more robust editing and better identity preservation.

In summary, our key contributions are:

- A framework for enhancing existing editing linear methods through self-conditional learning.
- An approach to editing that tackles shortcomings in the existing latent space by unfreezing the generator and fine-tuning it towards a fairer representation.

2. Related Work

Bias in Generative Networks Generative models have been extensively studied in the context of bias evaluation and mitigation. Early works focused on fairer generation with the aim of improving the performance of a downstream classifier [10, 15, 30, 42]. They employ importance re-weighting schemes [15], leverage a reference data set [10], or train a generator conditioned on the biased attributes [30, 42]. Others analyze the role of inductive biases in the generative process [45] or mitigate biases without training through better sampling of latent codes [34]. Yu et al. [44] studied Generative Bias in greater depth. They demonstrated that when training GANs, minority modes are more likely to suffer from collapse.

In contrast to these works, we investigate the bias as a root cause for poor performance in editing. We employ a self-conditioned model and a re-sampling scheme to mitigate this bias, stave off mode collapse, and achieve greater editability for minority attributes.

Latent-Space Editing The unprecedented ability of StyleGAN to encode semantic properties within its latent space has spawned an impressive array of image manipulation methods [16, 24, 32, 33, 40]. These methods typically aim to find linear directions in the latent space of the GAN, such that modifying a latent code along these directions will produce a change in a single semantic property of the generated image. These methods range widely in the level of supervision they require, from weak supervision in the form of binary attribute classification [32] through detailed 3D morphable face models [35]. Others have proposed ways to identify such semantic directions in an entirely unsupervised manner [16, 33] or in a zero-shot manner by leveraging models [26] that jointly encode image and text [24]. Linear editing directions, however, typically suffer from entanglement or rapidly deteriorating performance when applying large changes. Recently, it has been suggested that these shortcomings can be tackled by discovering non-linear paths in the latent space [3]. Typically these methods train a network to perform local manipulations on a given latent code [2, 20, 43]. Others suggest to model the warped manifold of the GAN [38] or traverse this manifold by finding a new local-basis at every step [9]. While such methods have enjoyed relative success, we argue that they aim at solving the problem by attacking the symptom rather than its cause. Instead of employing complex methods to find non-linear directions in the space of a pre-trained GAN, we propose to re-train the model and alleviate the bias which gives rise to these flaws.

Generator Unfreezing Typically, frameworks which utilize a pre-trained GAN for downstream tasks elect to keep the GAN frozen. Doing so is expected to bring stability and ensure higher quality. A recent line of works, however, challenges these assumptions and demonstrates that some tasks can be better handled by modifying the generator itself [7].

The intuition is that brief fine-tuning sessions, whether adversarial [41] or non adversarial [13], tend to keep the models well aligned and preserve most of the structure of the latent space. While this property is typically used for tasks that cannot be accomplished with the original generator, such as out-of-domain editing [13, 25], it has been shown that modifying the generator can also dramatically improve inversions [5, 23, 28], create new editing directions [8], improve temporal consistency [37] or enable a user to re-write the synthesis rules of the network [6].

In contrast to these works, we propose that generator fine-tuning can also be used to improve on existing editing approaches. By converting implicitly learned latent directions into explicit conditioning codes, we force the generator to contend with minority attributes, reduce the Generative Bias, and enable better control of the modified attributes under more significant modifications.

3. Method

To tackle the Generative Bias and enable more robust editing, we propose a scheme for converting a pre-trained generator and an existing linear editing direction into a self-conditioned model. We do so by tackling three aspects of
the training flow: First, we generate self-labeled data using distances in the latent-space of the generator. Second, we modify the generator’s architecture in order to explicitly represent a multitude of rare modalities. Third, we employ a re-sampled fine-tuning session with broadened discriminator supervision. These modifications are outlined below. An overview of the different steps is provided in Fig. 2.

3.1. Latent-Distances as Self-Conditional Labels

In a recent work, Nitzan et al. [22] demonstrated that the latent space distance of an image from a semantic editing hyperplane is linearly correlated with the magnitude of the semantic attribute in the image. Drawing on their insight, we propose to use these distances as labels for the original dataset. These labels could then be used to fine-tune the generator into an unbiased conditional model.

To self-label the data, we begin by extracting an editing hyperplane for each property that we want to de-bias. The typical method for identifying such planes is through the use of InterfaceGAN [32]. However, any method which extracts linear editing directions in any of the GAN’s multiple latent spaces is equally suitable. As we shall later demonstrate, our method can work equally well with directions extracted by a wide array of methods in multiple latent spaces, including: StyleCLIP [24] directions in \( S \) [40], InterFaceGAN [32] directions in \( W \), and SeFA [33] directions in \( W^+ \).

Armed with these editing directions, we next turn to labeling our training data. We invert all training set images into the latent space of the network using a pre-trained e4e [36] model. We then calculate, for each image, the latent space distance between its inverted code and the editing hyperplane. We find the minimal and maximal distances in the set and re-scale all distance labels such that the range of possible values is in \([-1, 1]\). We use these distances as a set of continuous labels for each image.

In the case of methods which produce editing directions without an intercept, we arbitrarily set the intercept to 0 (i.e. we use the projection of the latent code on the editing direction). As distances are then normalized, this choice bears no effect on the results.

3.2. Conditional Multi-Constants

Using the self-labeled data, we turn to converting our generator into a conditional model. Our goal is to enhance control and improve representation of rare dataset modalities. We thus adopt the multi-constant approach proposed by Sendik et al. [31].

Multi-constant models expand StyleGAN’s initial learned constant to a set of constants, each of which is expected to control a different modality present in the data. At inference time, a constant can be chosen either conditional on a label, or by augmenting StyleGAN’s mapping network to output a weight vector which denotes an importance-weighting score for each constant. One can then choose the most dominant constant, or simply mix them together in proportion to their weights. Sendik et al. [31] demonstrated that such a setup allows the model to better encode the unique attributes relevant to each modality, while allowing the rest of the network to share information from all modalities, leading to higher quality synthesis. In our case, the information sharing aspect is crucial as we are interested in better treatment of rare modalities, which may not contain sufficient samples to train a high quality GAN.

In practice, we modify the multi-constant approach in the following way: In addition to StyleGAN’s original constant \( C \), we add two new constants for each attribute we wish to control: \( c^+_i \), \( c^-_i \). The intuition here is that many attributes may not be symmetric around the average image (e.g. young faces differ significantly from old faces) and by introducing...
Figure 3. Overview of our multi-constant conditional generator. We treat the controlled attributes as different data modalities, introduced through modulations of the learned constant. Each modality has two additive constants, denoting a positive and negative attribute change with respect to the mean image. The direction and strength of the attributes is controlled through a vector $\mathbf{\alpha}$, where each entry $\alpha_i$ controls a different image property.

In contrast to Sendik et al. [31], our method uses additive constants for each attribute. This allows us to learn a model which can simultaneously control multiple attributes, without having to learn a constant for each attribute combination. Moreover, this motivates the constants to focus only the information which differs between the modalities, a simpler task which can be tackled with fewer data.

Typically, conditional StyleGAN models inject the conditioning code through the mapping network [17]. Adding such conditioning to a pre-trained model through fine-tuning is non trivial, owing to the increased dimensionality of the latent codes. In such a scenario, the mapping network has to be entirely replaced, or part of the latent code has to be re-purposed for the conditioning. Additive constants, meanwhile, can be used to modulate the existing network without any re-learning of the core generative path. In Sec. 4 we investigate the option of replacing the mapping network with a conditional one and find that this loss of information leads to poor editing control and demeand performance.

On the discriminator front, we employ the original StyleGAN2 [19] discriminator and augment it with a new prediction head tasked with regressing the mixture of modalities, $\alpha_i$, used in the image synthesis. In order to help the network encode differences between asymmetric modalities such as adding hats or glasses (where the ideal negative direction constant may simply be a tensor of zeros), we once again separate positive and negative editing directions. We do so by defining a 3-entry score vector for each attribute we want to control: $[|\alpha_i|, 1 - |\alpha_i|, 0]$ for $\alpha_i < 0$ and $[0, 1 - |\alpha_i|, |\alpha_i|]$ for $\alpha_i > 0$. We employ a soft cross-entropy loss between the discriminator’s predictions and these soft labels. For real images, we substitute $\alpha_i$ with the normalized distance-labels for each attribute in the image.

By combining these modifications, we create a conditional GAN architecture where conditioning codes, $\alpha_i$, are injected through linear interpolation of learned constants. Our discriminator is further incentivized to pay attention to rarer modalities, making mode collapse less likely. The full network architecture is outlined in Fig. 3.

### 3.3. Re-sampled Tuning

With the image labels and architecture modifications at hand, we turn to fine-tuning the GAN. We continue training using the original, now labeled, dataset. However, rather than sampling a random image at every iteration, we uniformly sample a random score for each attribute and draw the nearest-neighbor image in attribute space. By doing so, we further ensure that the discriminator sees an unbiased distribution.

### 3.4. Image Inversion

An essential requirement for latent space editing methods is the ability to modify real images, inverted into the latent space of the GAN [1, 4, 27]. As our method extracts particular semantics out of the latent space and into a set of explicit conditional labels, we find that codes in the original model are no longer tied to the same identities in the new model. We are thus unable to employ off-the-shelf encoders for use with
our model. However, such models can be easily adapted to our conditional setting. We do so by fine-tuning a pre-trained e4e [36] model. The encoder’s function is unchanged. It receives an image, and produces a latent code in \( \mathcal{W}^+ \). This code is then fed into our conditional generator, along with a set of latent-space distance labels derived from the same input image in the original source model. These distances can be efficiently approximated by simply passing the image through the discriminator’s mode prediction head. The encoder is fine-tuned using the same optimization goal and loss terms as in the original e4e model. A similar approach can be used to integrate our model with PTI [28], enabling accurate and highly editable reconstructions of real images.

3.5. Training Details

We train our models using the StyleGAN2-Pytorch implementation. We use the official FFHQ [18] 1024x1024 and AFHQ-Cat [11] 512x512 checkpoints. FFHQ models were fine-tuned using 30k iterations and a batch size of 4. The AFHQ model was fine tuned with 30k iterations and a batch size of 8. e4e models were fine-tuned using 50k iterations and a batch size of 2. When integrating with PTI, we use the latent provided by our fine-tuned e4e model as a pivot. The generator optimization is performed over 350 iterations. Learning rates, relative loss weights and all other parameters were left unmodified from the original models.

4. Experiments

4.1. Qualitative Comparison

We begin with a qualitative comparison of our method to existing editing techniques. In Fig. 4 we show sequential editing of synthesized images using our method. Our model is capable of successfully controlling multiple attributes, across large modifications, without significantly compromising the identity.

In Fig. 5 we compare our editing capabilities on real images to a range of linear editing methods. Real images were inverted into our model and into the official FFHQ 1024x1024 model, using PTI [28]. For each identity, we compare the editing performance of our model to the baseline linear editing direction which was used to extract the conditioning labels. Our method consistently outperforms these baselines. The performance gap widens as we move closer to the edges of the distribution, such as when considering large poses or old faces. When considering asymmetric attributes such as glasses, our model maintains the same image even as we keep moving in the no-glasses direction. The same operation with the StyleCLIP baseline, meanwhile, leads to an increase in age. These results demonstrate that editing performance can be improved, with no additional supervision, simply by addressing the Generative Bias and allowing the generator to be fine-tuned.

Finally, in Fig. 6 we compare our performance to non-linear alternatives: Local Basis [9] and StyleFlow [2]. Recall that our method does not maintain the identities of the original model. To facilitate comparisons in spite of this limitation, we use the alternatives to edit images synthesized by the original model, and project the same images into our models using PTI.

Despite the more challenging (inverted image) setup, our method still displays more robust editing performance, maintaining better identity through age changes and reducing deterioration for large poses. Importantly, unlike the supervised StyleFlow, our method relies only on the weak (left / right pose labels) or CLIP-based supervision used to find the initial editing directions (InterFaceGAN for pose, StyleCLIP for age).

As our experiments demonstrate, self-conditioned GANs are capable of consistent manipulations over larger spans than existing methods, even when compared to complex, non-linear approaches.
Figure 5. Linear editing comparisons on real images. In each pair of rows, we compare our method against the initial latent direction which was used to extract our self-conditioning labels. In all cases, our method better preserves the identity and allows for more significant manipulations before image quality suffers drastically. In the glasses example (bottom), continued movement in the negative direction on an image without glasses leads to an increase in age. Our model avoids this problem and produces the same identity.
4.2. Quantitative Comparisons

We quantitatively evaluate the performance of our method by considering identity preservation for large modifications.

We follow [28] and measure the cosine similarity between the embeddings of a pre-trained identity recognition network (ArcFace [12]) using pairs of pre- and post-editing images. When considering continuous attributes, we ensure similar magnitude of change by employing pre-trained pose [29] and age detection [39] networks. Images are edited until they reach a desired level of change, e.g., +20 years. For binary attributes, we observe that for any fixed step size, there exist a portion of images where the manipulation fails (e.g., glasses are not added), and a portion of images where the manipulation is too strong, and identity is lost. Increasing the step size leads to an increase in both successful manipulations, and in identity loss. To facilitate fairer comparisons, we thus test each method along a wide range of step sizes and report the identity preservation scores as a function of the percent of images that were successfully manipulated. A manipulation is considered successful if it causes an off-the-shelf classifier [18] to change its result.

The results are shown in Fig. 7. Our model maintains a higher degree of identity similarity for rare attributes (glasses) or in regions where data is sparse and the baseline generator tends towards mode-collapse (large age, pose), and performs on par with non-linear methods where data is abundant (smiles).

Figure 7. Identity preservation graphs. Our model maintains better identity for meaningful changes, and shows more moderate decline as we edit towards regions where training data was scarce. For non-minority attributes like smile, our model achieves comparable performance to non-linear editing methods, and outperforms the linear editing directions from which our self-labels were drawn.

4.3. Ablation study

We evaluate different aspects of our proposed method by conducting a qualitative ablation study. Specifically, we investigate our choice of a multi-constant setup, the use of latent-space labels, the importance of uniform sampling, and the benefits of unfreezing the generator.

In the first scenario, rather than spreading our modalities across constants, we build on the conditional setup of [17] and extend the latent code with a conditioning code that consists of one entry per property, with values $\alpha_i \in [-1, 1]$. For a model with control over $n$ attributes, the latent code therefore takes the form $\mathbf{z}' = \mathbf{z} \oplus \alpha_1 \oplus \alpha_2 \oplus \ldots \oplus \alpha_n$. The discriminator is similarly tasked with predicting the set $\{\alpha_i\}_{i=1}^n$. For real images, the $\alpha$ values are given by the self-labeling scores.
and more commonly to severe quality deterioration. Through constant interpolations lead to minor changes at best, the changes in constants. Attempts to manipulate the image these regions.

This is similar to that of our full model, but the quality of results near the edges of the distribution deteriorates, demonstrating our network ‘s tendency towards mode collapse around minority modalities – we observe that in some cases it can increase the network’s susceptibility to the other kind of bias which affects the network - entanglement of attributes in the dataset. However, in many cases this effect can be alleviated by training a multi-attribute model to further condition on the entangled attribute.

Finally, in the third scenario, we fine-tune the model with images sampled randomly from the FFHQ set, with no regard to their labels.

Finally, in the fourth scenario, we learn new constants without modifying any of the pre-trained generator’s weights.

In Fig. 8 we show the results of each experiment. When replacing the constants with a conditional mapping network, we observe a severely decreased range of control. We hypothesize that, as the mapping network needs to be trained from scratch, the optimization prefers to devote most efforts to re-aligning the generated distribution rather than to the attribute control. If this is the case, a change of hyper-parameters might enable successful conditional editing, but we were unable to find such parameters.

Using classifier scores similarly leads to a reduced editing range. This is likely a result of the quick saturation in classifier scores.

When we do not perform uniform sampling, editing range is similar to that of our full model, but the quality of results near the edges of the distribution deteriorates, demonstrating that the data bias contributes directly to the performance in these regions.

Lastly, if the generator remains frozen, it fails to adapt to the changes in constants. Attempts to manipulate the image through constant interpolations lead to minor changes at best, and more commonly to severe quality deterioration.

5. Discussion

We presented a self-conditioned generative model, designed to tackle the inherent Generative Bias of GANs and improve image editing. Our method leverages existing linear latent traversal methods, and empowers them to properly deal with minority modalities. In doing so, it achieves better identity preservation and direct control over image attributes.

Our results demonstrate the benefits of keeping fairness considerations in mind when dealing with generative tasks. More importantly, they provide further proof that there exist venues through which these, often painful, biases can be mitigated — without having to collect additional, unbiased data.

While our network demonstrated improved performance in regions that suffer from the Generative Bias – i.e., the GAN’s tendency towards mode collapse around minority modalities – we observe that in some cases it can increase the network’s susceptibility to the other kind of bias which affects the network - entanglement of attributes in the dataset. However, in many cases this effect can be alleviated by training a multi-attribute model to further condition on the entangled attribute.

In the future, we would like to further develop mechanisms that better exploit the existing data in order to promote better control over specific attributes of interest. We hope that our work serves as additional motivation for further explorations into fairer generative models, as these can impact both ethical and practical concerns.

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] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, 2020.
[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. ACM Trans. Graph., 40(4), 2021.
[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. arXiv preprint arXiv:2104.02699, 2021.
[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing, 2021.
[6] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model in regions that suffer from the Generative Bias – i.e., the GAN’s tendency towards mode collapse around minority modalities – we observe that in some cases it can increase the network’s susceptibility to the other kind of bias which affects the network - entanglement of attributes in the dataset. However, in many cases this effect can be alleviated by training a multi-attribute model to further condition on the entangled attribute.

In the future, we would like to further develop mechanisms that better exploit the existing data in order to promote better control over specific attributes of interest. We hope that our work serves as additional motivation for further explorations into fairer generative models, as these can impact both ethical and practical concerns.
model. In *European Conference on Computer Vision*, pages 351–369. Springer, 2020. 2

[7] Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the latent space of generative networks. In *International Conference on Machine Learning*, pages 600–609. PMLR, 2018. 2

[8] 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. 2

[9] Jaewoong Choi, Junho Lee, Changyeon Yoon, Jung Ho Park, Geonho Hwang, and Myungjoo Kang. Do not escape from the manifold: Discovering the local coordinates on the latent space of gans, 2021. 2, 5, 7

[10] Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, pages 1887–1898. PMLR, 2020. 2

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[13] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators, 2021. 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[15] Aditya Grover, Jiaming Song, Alekh Agarwal, Kenneth Tran, Ashish Kapoor, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, 2019. 2

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 4, 7

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 5, 7

[19] 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. 4

[20] Bingchuan Li, Shaofei Cai, Wei Liu, Peng Zhang, Miao Hua, Qian He, and Zili Yi. Dystyle: Dynamic neural network for multi-attribute-conditioned style editing, 2021. 2

[21] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14286–14295, 2020. 1

[22] Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. Large: Latent-based regression through gan semantics, 2021. 2, 3

[23] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *European Conference on Computer Vision*, pages 262–277. Springer, 2020. 2

[24] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021. 1, 2, 3

[25] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 2

[26] 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. *arXiv preprint arXiv:2103.00020*, 2021. 2

[27] 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. *arXiv preprint arXiv:2008.00951*, 2020. 4

[28] 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. *arXiv preprint arXiv:2008.00951*, 2020. 4

[29] Nataaniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 7
[30] Prasanna Sattigeri, Samuel C Hoffman, Vijil Chentharamarakan, and Kush R Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3–1, 2019. 2

[31] Omry Sendik, Dani Lischinski, and Daniel Cohen-Or. Unsupervised k-modal styled content generation. *ACM Transactions on Graphics (TOG)*, 39(4):100–1, 2020. 3, 4

[32] Yujun Shen, Jinjin Gu, Xiaou 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. 1, 2, 3

[33] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 2, 3

[34] Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020. 2

[35] Ayush Tewari, Mohamed A. Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Perez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6141–6150, 2020. 2

[36] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021. 3, 5

[37] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H. Bermano, and Daniel Cohen-Or. Stitch it in time: GAN-based facial editing of real videos, 2022. 2

[38] Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. WarpedGANSpace: Finding non-linear rbf paths in GAN latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6393–6402, October 2021. 2

[39] Yusuke Uchida. Age estimation - pytorch, 2018. [https://github.com/yu4u/age-estimation-pytorch](https://github.com/yu4u/age-estimation-pytorch). 7

[40] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 2, 3

[41] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models, 2021. 2

[42] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE, 2018. 2

[43] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. *2021 International Conference on Computer Vision*, 2021. 2

[44] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision*, pages 377–393. Springer, 2020. 1, 2

[45] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jianming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *arXiv preprint arXiv:1811.03259*, 2018. 2