Delta-GAN-Encoder: Encoding Semantic Changes for Explicit Image Editing, using Few Synthetic Samples.

Nir Diamant  Nitsan Sandor  Alex M. Bronstein

Technion - Israel Institute of Technology
{nirdiamant21,nits210}@gmail.com

Abstract

Understating and controlling generative models’ latent space is a complex task. In this paper, we propose a novel method for learning to control any desired attribute in a pre-trained GAN’s latent space, for the purpose of editing synthesized and real-world data samples accordingly. We perform Sim2Real learning, relying on minimal samples to achieve an unlimited amount of continuous precise edits. We present an Autoencoder-based model that learns to encode the semantics of changes between images as a basis for editing new samples later on, achieving precise desired results - example shown in Fig. 1. While previous editing methods rely on a known structure of latent spaces (e.g., linearity of some semantics in StyleGAN), our method inherently does not require any structural constraints. We demonstrate our method in the domain of facial imagery: editing different expressions, poses, and lighting attributes, achieving state-of-the-art results.

1. Introduction

Over the last few years, the concept of Generative Adversarial Networks (GANs) has been developing rapidly and improved the ability to synthesize more photo-realistic data samples, specifically images. Even though recent GANs’ architectures are designed to have semantic latent space nature (e.g., StyleGAN variants [18, 19, 17]), explicit control over the latent space has not yet been achieved. StyleGAN’s mapping network $f(z)$ that maps the input latent vector $z$ to an intermediate latent vector $W$ contributes to disentangling the $W$-space, since it does not have to match the probability density of the training data as $z$. Additionally, a disentangled representation will ease the generation of realistic samples [18, 29, 16, 32]. However, the different $W$-space semantics are encoded in an imperfectly linearly-separated manner, introducing difficulties in generating a specific desired output.

CGANs [23] can overcome the problem of attribute entanglement, since it is trained to separate outputs. Despite
this, CGANs are limited to features chosen pre-training. Another major drawback to achieving feature disentangled results is the imprecise control over the GAN output: the lack of an accurate metric for distinguishing between any two possible attributes prohibits the generation of every specific pre-defined image. The existence of a real-world derived dataset containing the same samples differing only by specific attributes may contribute to the definition and inference of exact attributes, though this is nearly impossible to achieve.

Studies have been conducted on how to project real-world images into their matching latent vectors to learn and control the behavior of latent spaces. For StyleGAN, it is common to project an image into $W^+$ vectors [1, 2], which are concatenation of 18 different $W$ vectors. Nevertheless, projections into a GAN’s latent space still cannot mimic the image generation’s inverse function, and exhibit a known issue of the trade-off between qualitative image reconstruction and the latent vector’s semantics.

Generative Normalizing Flows [21, 20, 14, 15] can perfectly project an image by calculating the inverse of the generative function thanks to their property of reversibility, since they are composed of a series of bijective functions. A significant disadvantage is that the latent vector size must match the image dimensions, producing a high-dimensional latent space whose exploration time is longer by orders of magnitude.

Recent work by Schwartz et al. [28] proposes a method of encoding changes between images in the feature space using an Autoencoder-based model. The encoded changes are then used for augmenting sparse classes and improving classification accuracy.

Deriving from Schwartz et al. [28], we propose the idea of encoding the changes ($\Delta$-s) and put forth a method that enhances this concept for generating accurate edited qualitative data samples using a pre-trained GAN. To achieve this goal, we present a new method requiring three crucial elements: a qualitative projection method (i.e., reconstruction and editability), multiple samples differing by only a single attribute, and a model that can encode the changes and apply them to edit latent vectors representing images.

Our contribution is in providing a method that:

1. Allows the editing of images with unlimited specificity unencumbered by attribute entanglement, learning to isolate the different semantics.

2. Learns the behavior of the semantics without the requirement of any structural constraints, allowing it to find the precise nonlinear path of the edit in the latent space.

Our code and data will be available to download from www.xxxxxxx.com.

2. Related Work

Different projects have been completed towards achieving the disentanglement of features in generative models [22, 10, 34, 25]. Shoshan et al. [31] proposed using pre-trained task-oriented models to reward and penalize every two samples that share or differ by specific criteria, thus clustering the semantics within the latent space structure. Shen et al. [30] added a face classifier as a third player to compete against the Generator in order to sharpen its ability to preserve the identity of the generated image.

Learning from synthetic data and combining 3D models with GANs were used for multi-domain applications (e.g., expression transfer, pose changing, 3D face out of 2D face constructing, face de-occlusion, face frontalization, face decoupling, etc.) [12, 24, 11, 38, 37, 4].

Some works have leveraged the relatively linear separable structure of StyleGAN’s latent space and explored the semantics of linear differences between latent vectors [1, 2]. Other studies were conducted to find meaningful linear paths in the latent space [29, 36, 13]. Zhu et al. [40] studied where good latent codes are located in the latent space.

Another related task is image projection into the latent space for editing real image samples using generative models. Some methods use iterative optimization algorithms [19, 7], while other methods use different kinds of Encoder-based architectures to project an image into its matching latent vector. The latter tend to reconstruct less accurately yet significantly more quickly, embedding the vector much more semantically. [39, 26, 35].

Tov et al. [33] designed an encoder that embeds images into semantically better latent vectors with a slight reconstruction quality trade-off. Alaluf et al. [6] enhanced both encoders from [26, 33] by using their models iteratively and learning the residual with respect to the current estimate of the latent vector at each step until convergence. Alaluf et al. [6] studied the latent space semantics of aging transformation, supervised by an age prediction model, and showed that the actual latent path is nonlinear, surpassing the results of a linear path. Abdal et al. [3] formulated conditional exploration in the latent space as an instance of conditional continuous normalizing flows to enhance attribute-conditioned sampling and attribute-controlled editing. They showed that the nonlinear paths surpass the quality of the results that correspond to the linear path.

3. Method

3.1. General Idea

Our method finds the latent vector $\mathbf{a}_{ij} \in W^+$ in a pre-trained GAN that matches the desired transformed image $\mathbf{A}_{ij}$, given an input image $\mathbf{A}_i$. It does it by first learning $\Delta_{i,j}$, a low dimensional vector that represents a semantic
change defined by two images, and then combines it to the input vector \( a_i \in W^+ \) to get the desired result. Both parts are learned together during the training phase of our model. Full model architecture is illustrated in Fig. 3.

We assume that using projection methods that embed images semantically into \( W^+ \) \cite{Karras2020-a} vectors will embed less-realistic images semantically similar to embedding real ones (e.g., the manipulation over the latent vector for editing the eyes of a natural person is similar to editing the eyes of a synthetic face). This vital observation allows us to learn the task on synthetic data and infer the results over real-world samples.

To supervise the process of learning the \( \Delta \)-s, we create a small image dataset generated from synthetic 3D models of one female and one male - an example is shown in Fig. 2. The dataset consists of two classes, \( A \) and \( B \), for each we have n-sized sequence of samples, \( A_1, A_2, ..., A_n \) and \( B_1, B_2, ..., B_n \). In each series, one of its attributes changes along the sequence, while all the other attributes remain untouched. We then project all the \( A_i \) and \( B_i \) samples into their matching latent vectors \( a_i, b_i \) according to the related pre-trained GAN.

### 3.2. Model Architecture and Loss Function

Our model is based on an Autoencoder architecture, where the Encoder \( E \) receives two latent vectors, \( a_i, a_j \in W^+ \), that match the images \( A_i, A_j \). Both images belong to the same class (denoted alphabetically) and differ by some attribute (denoted by subscript). The model encodes them to \( \Delta_{i,j} \), a lower-dimensional vector that represents the semantic distinction between the inputs.

\[
\Delta_{i,j} = E(a_i, a_j). \tag{1}
\]

We denote the space, consisting of all such \( \Delta \)-s as \( \Delta \)-space. We enforce linear space properties on the \( \Delta \)-space as described in subsection 3.4.

The Decoder \( D \) receives a latent vector \( b_i \), that matches the image \( B_i \), and \( \Delta_{i,j} \). It produces an intermediate term \( b_{\text{residual}} \) which is added to \( b_i \) to encourage the model to learn the residual of \( b_j \) and \( b_i \). \( D \) produces the residual \( b_{\text{residual}} \) instead of \( b_j \), since it is an easier task than learning the explicit output directly.

\[
b_j = D(b_i, \Delta_{i,j}) + b_i. \tag{2}
\]

The process is done in a supervised manner using two losses: (1) Identity loss - the \( \Delta \) is defined and applied to the same class. (2) Transfer loss - the \( \Delta \) is defined by one class and applied to another.

\[
l_{\text{residual}} = \lambda_1 \cdot \|a_j - \hat{a}_j\|^2 + \lambda_2 \cdot \|b_j - \hat{b}_j\|^2. \tag{3}
\]

The above explanation holds where the classes of \( A, B \) and the indices \( i,j \) change roles during training. This mechanism encourages the model to be class symmetric and emphasizes that the order of inputs to \( E \) is important: opposite order of inputs, \( (j,i) \) instead of \( (i,j) \), yields an exact opposite \( \Delta_{j,i} \), a linear property that we force over the \( \Delta \)-space.

### 3.3. Real Time Data Augmentation

To avoid over-fitting, we enrich the data by adding random noise \( n \sim \mathcal{N}(0, \sigma^2) \) to every latent vector that belongs to the same class during the training phase.

\[
\begin{align*}
& \hat{a}_i \leftarrow a_i + n_a \\
& \hat{b}_i \leftarrow b_i + n_b.
\end{align*} \tag{4}
\]

Since tiny latent space perturbations in StyleGAN change the semantics in a disentangled manner, adding the same relatively small noise to different latent vectors in \( W^+ \) preserves semantic changes between them. On the other hand, above a certain magnitude, a noise might transfer the latent vectors to other areas in the latent space that do not maintain the exact change of semantics. Examples shown in Fig. 4.

Effectively, each attribute’s \( \Delta \) is learned by an unlimited number of samples, instead of only one female and one male. An equation that describes a full pipeline of the model would be:

\[
\hat{b}_j = D(E(a_i + n_a, a_j + n_a), b_i + n_b) + b_i. \tag{5}
\]

### 3.4. Enforcing Linear Space Properties

To enforce the model learning a relative \( \Delta \) \( \Delta_{i,j} \), instead of attribute transfer, we force the \( \Delta \)-space to be linear and orthonormal.

We do it during the training phase by decoding the same \( \Delta_{i,j} \) with all the latent vectors that belong to the same attribute’s series \( a_i, a_{i+1}, ..., a_{n-1-(j-i)} \), adding \( \alpha \cdot \Delta_{i,j} \), and demanding the matching endpoints. To enforce the linear space properties, the parameter \( \alpha \) is a scalar set according to the following equation:

\[
D(a_k, \alpha \cdot \Delta_{i,j}) = \hat{a}_k + \alpha \cdot (j-i). \tag{6}
\]
Figure 3: Our model’s structure: Two images \( \mathbf{A}_i, \mathbf{A}_j \) are projected into the GAN’s latent vectors \( \mathbf{a}_i, \mathbf{a}_j \in W^+ \) with a semantic projection method \( G^{-1} \). Both latent vectors are then fed into the encoder \( E \) to calculate a small vector, \( \Delta_{i,j} \), representing the directed semantic change between \( \mathbf{a}_j \) and \( \mathbf{a}_i \). \( \Delta_{i,j} \) is being fed into the decoder \( D \) once with \( \mathbf{a}_i \) and then with \( \mathbf{b}_j \) which outputs the residual needed to get \( \mathbf{a}_j \) and \( \mathbf{b}_j \) respectively. The whole pipeline is symmetric for \( \Delta_{j,i} \).

Figure 4: Data augmentation by adding noise to the latent vectors: A - the original samples before and after the change \( \mathbf{a}_i, \mathbf{a}_j \). B - small-amplitude noise \( (\sigma^2 = 1, \sigma^2 = 2) \) is added to the latent vectors such that the same \( \Delta \) between the images is preserved with no additional relative changes. C - Too much noise is added \( (\sigma^2 = 4, \sigma^2 = 5) \), making the images differ by more attributes than desired (i.e., the left pair is also differentiated by the amount of hair; the right pair by opening the mouth).

Figure 5: Enforcing linear behavior in the \( \Delta \)-space: The left ellipse demonstrates the behavior in the GAN’s \( W^+ \) space; the paths of two series of the same attribute changing along a scale of two different classes are shown. The paths are different, nonlinear, and not scaled. The right ellipse demonstrates the transformation enforced in the \( \Delta \)-space: the difference between two consecutive images, as represented in the \( \Delta \)-space, is scaled and oriented similarly, and is uniform among the different classes. Also, the addition of consecutive \( \Delta \)-s constructs the cumulative \( \Delta \).

An illustration of the linearity constraint is shown in Fig. 5.

Our model learns the \( \Delta \)-s between the images with two biases: (1) A bias caused by the distribution difference between the synthetic samples and the real images. (2) A bias caused by the imperfect projection method in terms of reconstruction and editability. To mitigate these issues, we validate the editability of projected real images and,
4. Experiments

4.1. Data Analysis

First, to observe the relation between the synthetic data and images of real faces, we plot a low dimension distribution of the projection for both of them. The projection shows that the synthetic data is separated from the distribution of the real-world images. Nevertheless, linear interpolation of the differences corresponding to the same attribute change fall inside the same distribution, thus allowing studying from synthetic data for later inferring over real-world samples (Fig. 6).

Even more importantly, the editability of random images the GAN synthesized, since the latent space semantics are best reflected on them.

4.2. Comparison to Linear Paths

We show a series of comparisons to the linear paths edits that matches the state-of-the-art projection of ReStyle [6] based on PSP [26]. We compare editing different attributes to show the comprehensive ability of the technique: different expressions that combine several changes (e.g., emotions like happy or sad), and expressions that change only a single attribute (e.g., winking). We also test our model over changes in pose and lighting. Our results outperform state-of-the-art in terms of FID score (realism of the edited outputs) in Table 1, and in terms of cosine similarity between the output images’ features along with sequences of editing (the edited outputs reliably preserve the identity) in Table 2.

Above each comparison, we show the desired change defined by three images from two synthetic classes in order to demonstrate the input to the model (Fig. 7). The comparison is performed between random latent vectors from the GAN’s $W^+$ distribution, emphasizing that our model learns the latent space specific semantics despite the different biases mentioned in subsection 3.4.

We then show editing of different attributes on images of real people projected to $W^+$ vectors by the same manner we had projected our synthetic models (Fig. 8).

To assure the nonlinearity of the $\Delta$-s, we present graphs representing the nonlinear paths in the GAN’s latent space that correspond to the edits for each attribute family (expression, pose, lighting) as shown in Fig. 9.

4.3. Unsupervised Precise Pose Control

Another benefit of our Sim2Real model is controlling the exact degree of posture change over randomly synthesized (Fig. 10) and real-world images (Fig. 11). Thus, we achieve supervised behavior over an unsupervised GAN in terms of pose control. We measure the output face pose degree using [27], and reach mean degree error of $0.15^\circ$ and standard deviation of $3.37^\circ$ over changes in range $[-30^\circ, 30^\circ]$ (Fig. 12).

4.4. Method Limitations

Lastly, we show several cases where our model fails to achieve its exact goal, resulting in some entangled attribute change. There might be several reasons for failing to achieve the goal image:

1. Imperfect projection of images into the latent space in terms of semantic meaning.

2. Uncommonness of expressions in the GAN’s training data.

The failure cases might indicate that some attributes demand more strenuous effort to disentangle than others - as shown in Fig. 13.
(a) Expression change with only one detail change: Closing eyes.

(b) Expression change with several details together: Being happy.

(c) Expression change with several details together: Being sad.

(d) Pose change: Head right.

(e) Pose change: Head back.

(f) Light change: lighting is approaching from left to right.

Figure 7: Demonstration of our model’s results comparing to the linear path that defines the semantic difference in Style-GAN2. The first two rows are the input to the model, where the 3rd row corresponds to our model’s results, and the last row corresponds to linear editing. The different $\Delta$-s corresponds to a single detail, several details, pose or lighting changes.

4.5. Implementation Details

Our Auto-encoder-based model is constructed of an almost-symmetric Encoder and Decoder. The Encoder’s first layer size is twice the size of a $W^+$ vector flattened, followed by three hidden layers at the size of a pre-defined $\Delta \in \mathbb{R}^{64 \times 1}$ and another last output layer at the size of 64.

The Decoder receives an input sized ($\Delta \in \mathbb{R}^{64 \times 1} + W^+ \in \mathbb{R}^{512 \times 18}$), followed by the same sized amount of hidden layers and the last layer of a $W^+ \in \mathbb{R}^{512 \times 18}$ vector-sized vector, for outputting the edited residual vector. All intermediate layers are followed by a Leaky-Relu activation layer with $p = 0.25$ and a Dropout layer with $p = 0.2$. The model is trained with an ADAM optimizer with a learning rate of 1e-4 and weight decay of 1e-5. We find that after 10k-20k epochs, which last about 5-10 minutes, the model converges, depending on the specific task, on a single RTX 2080 GPU.

We measure the FID distance with Pytorch-fid and calculate the Cosine Similarity score over the features of the images extracted with a pre-trained ARCFACE model [8].

Our synthetic data is facial action coding system (FACS) based [9], and projected to latent vectors with ReStyle-encoder [6] based on the PSP method. [26].
5. Conclusions and Future Research

With only a few samples, which differ by a specific attribute, one can learn the disentangled behavior of a pre-trained entangled generative model. There is no need for exact real-world samples to reach that goal, which is not necessarily feasible. By using non-realistic data samples, the same goal can be achieved thanks to leveraging the semantics of the encoded latent vectors. Applying wanted changes over existing data samples can be done with no explicit latent space behavior exploration.

Future research may include testing the limits of the distance between the real world and simulation data, e.g., using a sketched-like simulation, or learning the function that maps the distribution of the synthetic data into the distribution of the real-world data. The concept may not be exclusive to facial images or even images in general, e.g., illustrated rooms can form a dataset to train a model to change specific attributes like furniture, objects, room structure, and more. The method proposed throughout this work serves as a sound basis for expansion into the investigation of broader applications.
\[ \Delta = 0^\circ \quad \Delta = 10^\circ \quad \Delta = 20^\circ \]

Figure 10: Generation of images with precise \( \Delta \)-s of head pose angles, without any pre-trained GAN supervision.

\[ \Delta = 0^\circ \quad \Delta = 10^\circ \quad \Delta = 20^\circ \]

Figure 11: Applying precise \( \Delta \)-s to projections of real people images of head pose angles, without any pre-trained GAN supervision.

Figure 12: Density graph of angles error across all different angles \( \Delta \)-s in the range of \([-30^\circ, 30^\circ]\). mean error of 0.15\(^\circ\), and standard deviation of 3.37\(^\circ\).

Figure 13: Failure cases of our method.

(a) Additional unwanted changes: Hair growth and eye opening.

(b) Additional unwanted changes: Hair style, mouth posture change, eye color and structure change.

(c) Additional unwanted changes: Hair color, face hue, face width.
References

[1] Rumeen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space?, 2019. 2, 3
[2] Rumeen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images?, 2020. 2, 3
[3] Rumeen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, 2020. 2
[4] Victoria Fernandez Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, and Edmond Boyer. A decoupled 3d facial shape model by adversarial training, 2019. 2
[5] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement, 2021. 2, 5, 6
[6] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement, 2021. 2, 5, 6
[7] Rushil Anirudh, Jayaraman J. Thiagarajan, Bhavya Kailkhura, and Timo Bremer. Mimicgan: Robust projection onto image manifolds with corruption mimicking, 2020. 2
[8] Juankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition, 2019. 6
[9] Paul Ekman and Erika L Rosenberg. What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA, 1997. 6
[10] Chanho Eom and Bumsuh Ham. Learning disentangled representation for robust person re-identification, 2019. 2
[11] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2019. 2
[12] Zhenglín Geng, Chen Cao, and Sergey Tulyakov. 3d guided fine-grained face manipulation, 2019. 2
[13] Erik Harkonen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls, 2020. 2
[14] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows, 2020. 2
[15] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows, 2019. 2
[16] Ali Jahanian, Lucy Chai, and Phillip Isola. On the “steerability” of generative adversarial networks, 2020. 1
[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020. 1
[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 1
[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. 1, 2
[20] Diederik P. Kingma and Prafulla Dharwal. Glow: Generative flow with invertible 1x1 convolutions, 2018. 2
[21] Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, page 1–1, 2020. 2
[22] Bingchen Liu, Yizhe Zhu, ZuoHui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization, 2020. 2
[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. 1
[24] Fania Mokhayeri, Kaveh Kamali, and Eric Granger. Cross-domain face synthesis using a controllable gan, 2019. 2
[25] Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Do generative models know disentanglement? contrastive learning is all you need, 2021. 2
[26] 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, 2021. 2, 5, 6
[27] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints, 2018. 5
[28] Eli Schwartz, Leonid Karlinsky, Joseph Shlom, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition, 2018. 2
[29] Yujun Shen, Jinjun Gu, Xiaou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing, 2020. 1, 2
[30] Yujun Shen, Ping Luo, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaoou Tang. Face2id-gan: Learning a symmetric three-player gan for identity-preserving face synthesis. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 821–830, 2018. 2
[31] Alon Shoshan, Nadav Bhounker, Igor Kviatkovsky, and Gerhard Medioni. Gan-control: Explicitly controllable gans, 2021. 2
[32] Ayush Tewari, Mohamed Elgharib, Gaurav Bhardwaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Riggng stylegan for 3d control over portrait images, 2020. 1
[33] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation, 2021. 2
[34] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1283–1292, 2017. 2
[35] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. A simple baseline for stylegan inversion, 2021. 2
[36] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation, 2020. 2
[37] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large- pose face frontalization in the wild, 2017. 2
[38] Xiaowei Yuan and In Kyu Park. Face de-occlusion using 3d morphable model and generative adversarial network, 2019.
[39] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing, 2020.

[40] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2021.