FADER NETWORKS: A HEURISTIC APPROACH

Marcos Cardenas Zelaya, Marie Leech, Viet Nguyen

April 18, 2019

ABSTRACT

In recent years, approaches to approximating complex data distributions have been centered around the generative adversarial networks (GANs) paradigm, eliminating the need for Markov chains in generative stochastic networks or approximate inference in Boltzmann machines. Applying GANs to image and video editing have been done in a supervised setting where external information about the data allows the re-generation of real images with deterministic complex modifications, using Invertible conditional GANs (IcGANs). Fader Networks extend on this idea by learning a post-encoding latent space invariant to labeled features on the image, and re-generating the original image by providing the decoder with alternate attributes of choice. In this paper, we explore the impacts of modifications on the encoding and decoding convolutional blocks, analyzing the effects of droput in the discriminator, implementations of different loss functions on the generated images' quality using appropriate metrics and extend the model by including Skip Connects. We finish by providing an empirical assessment on how Fader networks develop a pseudo-understanding of higher-level image features.

1 Introduction

Lample et al.'s Fader networks (FadNets) [11] are an encoder-decoder neural network that incorporates a discriminator network along with an adversarial loss[5] to learn a latent representation space invariant to the labeled features. For instance, an image of the same person with or without glasses, smiling or not smiling, would have in principle the same latent representation. Their aim is achieving the ability to manipulate certain attributes of interest of images. In the past, scholars have had attempts at this challenge (and more generally, unsupervised domain transfer) using various models [24, 18, 22] with and without generative networks. However, as Lample et al. pointed out, training is usually unsupervised, and domain transformation is usually ill-defined in a sense that there are no examples of transformations (for instance, no real image of both a masculine and feminine version of a person). Approaches such as variational autoencoders (VAEs) [10] have demonstrated promising results, however as domain transformation happens on a pixel level, the model and its variants suffer from poor reconstruction quality (blurriness, reduced in color sharpness) and often times hurts the interpretability and naturalness of the images. This effect is further emphasized when models attempt to make relatively large changes on the image such as adding sunglasses or changing hairstyle, in a sense that the necessary change takes up a large portion of the image pixels. Moreover, an inherent difference between VAEs and GANs is that VAEs focus on a reconstruction loss whereas GANs rely on constructing realistic images. Fader networks provide a new approach to domain transformation as the model incorporates a discriminator layer on top of an auto-encoder architecture, and learns an invariant latent representation space, allowing explicit control on some attributes of interest.

2 Related Work

Image generation. At the heart of unsupervised image generation is the Generator Adversarial Networks (GANs), which use two neural networks, one as the discriminator and other as the generator, to iteratively improve the model by playing a minimax game [5]. GANs aims at generating images without specific attributes. Several methods build off the GAN architecture to produce images with specified attributes by conditioning the networks to emphasize certain features, allowing the network to learn representations of those features, a more targeted approach to generative learning. Example method includes infoGAN [2] which is able to learn distributed representations in an unsupervised

manner by disentangling writing styles from digit shapes on the MNIST dataset. InfoGAN is able to predict attributes and reproduce them, but is not able to transform images. Despite GANs image synthesis powers, the generators continue to operate as black boxes, the understanding of various aspects of the image synthesis process, such as the origin of stochastic features, is still lacking [1]. It is often difficult to automatically discover high-level attributes such as glasses, age, or gender in the GAN paradigm without introducing a certain degree of inductive bias over the training samples.

Conditional image generation. Conditional image generation is mid-way between completely supervised and unsupervised learning. It draws from the conditional generative model for learning to disentangle the hidden factors of variation within a set of labeled observations [15]. However, in [15], their framework can only generate the images rather than modifying an *existing* image based on attributes. Methods from unsupervised domain transfer can also be applied to this area in which one maps an image from one domain to the other without supervision [7, 12, 8]; this is relevant to our work as the domain would correspond to an attribute value. In this area, specifically applied to transforming attributes is the conditional GAN (cGAN) and its extension, the Invertible conditional GAN (IcGAN) [18]. The IcGAN trains a GAN where the introduction of external information allows it to determine specific representations of the generated images. Then it evaluates encoders to inverse the mapping of the cGAN, i.e., mapping a real image into a latent space and a conditional representation. The IcGAN can be used to reconstruct and modify real images of faces conditioning on specified attributes, just as our method aims to do. As such, it is used as a baseline in the FadNet paper. However, we do not include the IcGAN in our analysis as it it beyond the scope of an ablation study. Similar work done by Yan et al. [23] proposed an attribute-conditioned deep variational auto-encoder framework that enables image generation from visual attributes. Note however that Yan et al.'s model can not directly modify the attributes of images. Yin et al. [25] formulate a semi-latent facial attribute space that systematically learns the relationship between user-define and latent attributes. It is capable of transforming several attributes of an image at a time, however it is prohibitively computationally expensive given our current resources.

Adversarial training. The training criterion used in this method is derived from the work on learning invariant latent spaces using adversarial training in domain adaptation [4, 3] and robust inference [14]. In these works, the end goal is to filter out nuisance variables, however we require the opposite. That is, we learn generative models and invariance is used to force the decoder to use attributes in its reconstruction.

3 Fader Networks

Let X and Y be the training set of images and associated attributes. Y can be any binary attribute of a face such as glasses/no glasses, young/old, mouth open/mouth closed, moustache/no moustache, etc. However, for simplicity, in this paper Y will be the binary attribute of whether the image is of a male or female. We thus have a training set $D = (x^1, y^1), ..., (x^m, y^m)$, of m pairs (image, gender) $(x^i \in X, y^i \in Y)$. The Fader networks goal is to learn from D a model that will generate an output (x', y') from an input (x, y), where x' is a generated version of input image x whose attribute y has been "transformed" into y'.

Encoder-decoder architecture. The fader network is based on an encoder-decoder architecture with domain-adversarial training on the latent space (see Figure (1) for an outline of the model). The encoder, $E_{\theta_{enc}} : X \to \mathbb{R}^N$, is a convolutional neural network with parameters θ_{enc} that maps an input image x to its N-dimensional latent representation $E_{\theta_{enc}(x)}$. The decoder, $D_{\theta_{dec}}$ is a deconvolutional network with parameters θ_{dec} that produces a new version of the input image given its latent representation $E_{\theta_{enc}}(x)$ and any attribute vector y'. In the Fader Networks paper the auto-encoding loss function used is the mean square error, MSE, that measures the quality of the reconstruction at the pixel level of an input x given its true attribute vector y:

$$\mathcal{L}_{AE}(\theta_{enc}, \theta_{dec}) = \frac{1}{m} \sum_{(x,y) \in D} ||D_{\theta_{dec}}(E_{\theta_{enc}}(x), y) - x||_2^2 \tag{1}$$

The purpose of the encoder-decoder is that modifying y in $D_{\theta_{dec}}(E_{\theta_{enc}}(x), y)$ generates images with different targeted attributes, but everything else in the image x will remain the same. In this simplistic architecture, the decoder will learn to ignore the attributes and thus have no effects on y on a test set. To alleviate this problem, the authors' proposed approach is to learn latent representations that are invariant with respect to the attributes. Simply put, given two versions of the same image x and x' that differ only in their attribute value, the two latent representations $E(x)$ and $E(x')$ should be the same as well. When this invariance is satisfied, the decoder uses the attribute to reconstruct the original image. As this cannot be easily added to the loss function, the authors propose using adversarial training on the latent space so as to incorporate this constraint into the loss function.

This "constraint", the *discriminator*, is an additional neural network that is trained to identify the true attributes y of a training pair (x, y) given $E_{\theta_{enc}}(x)$. The authors explain that the invariance is obtained by learning on the encoder

$E_{\theta_{enc}}$ such that the discriminator is unable to identify the true attribute y. Similar to generative adversarial networks, GAN's, this corresponds to a two-player game in which, informally, the discriminator tries to distinguish whether a feature y is in the encoded space $E_{\theta_{enc}}(x)$ or not, while the encoder tries to fool the discriminator.

Discriminator objective. The discriminators objective is to determine the true attribute y of a training pair (x, y). It outputs the probabilities of an attribute vector, $P_{\theta_{dis}}(y|E_{\theta_{enc}}(x))$, where θ_{dis} is the discriminators parameters. Its loss depends on the current state of the encoder as:

$$\mathcal{L}_{dis}(\theta_{dis}|\theta_{enc}) = -\frac{1}{m}\sum_{(x,y)\in D} logP_{\theta_{dis}}(y|E_{\theta_{enc}}(x)) \tag{2}$$

As the discriminator tries to determine whether a feature y is in the encoded space or not, while the encoder tries to fool the discriminator, this process leads to the removal of the feature y from the $E_{\theta_{enc}}(x)$ by the encoder. The encoded feature $E_{\theta_{enc}}(x)$ therefore does not have any information of y. However, since the decoder needs to reconstruct the same input image, $E_{\theta_{enc}}(x)$ has to maintain all information, except y and the decoder should get the feauture y from the input of the decoder.

Adversarial objective. The objective of the encoder is now two fold; to "fool" the discriminator from predicting y given $E_{\theta_{enc}}(x)$, while also providing enough information so that the decoder can reconstruct the image x given $E_{\theta_{enc}}(x)$ and y. Formally, the objective of the encoder is to compute a latent representation that optimizes these two objectives. The discriminator makes a mistake when it predicts 1-y. Given the discriminator's parameters, the loss of the encoder-decoder is now:

$$\mathcal{L}(\theta_{enc}, \theta_{dec}|\theta_{dis}) = \frac{1}{m}\sum_{(x,y)\in D} ||D_{\theta_{dec}}(E_{\theta_{enc}}(x), y) - x||_2^2 - \lambda_E logP_{\theta_{dis}}(1-y|E_{\theta_{enc}}(x)) \tag{3}$$

where $\lambda_E > 0$ controls the trade-off between the quality of the reconstruction and the invariance of the latent representations. Small values of λ_E will limit the decoder's dependency on the latent code y and will result in poor effects when altering attributes, while large values will restrain the amount of information of x contained in $E_{\theta_{enc}}(x)$ and will result with blurry images.

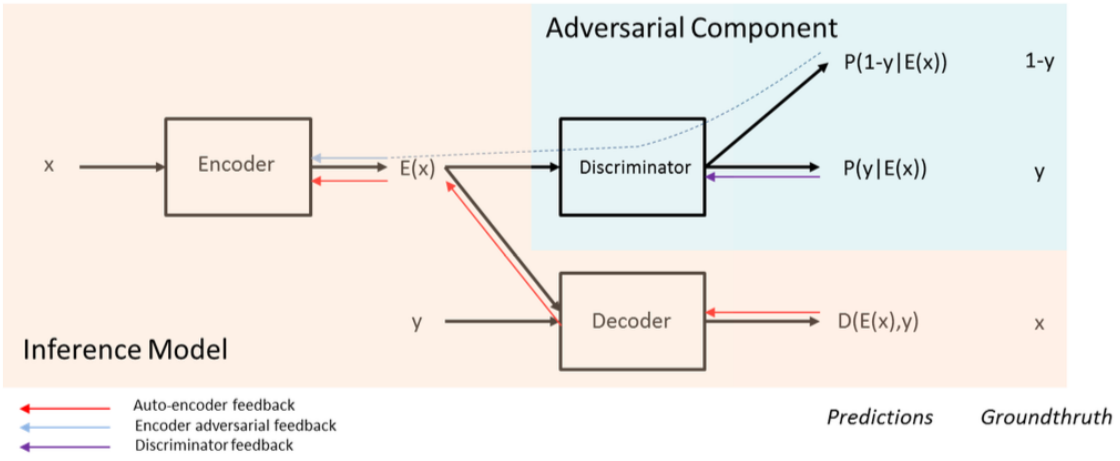

Figure 1: Main architecture. An image and attribute pair (x, y) is given as input. The encoder maps x to the latent representation z; the discriminator is trained to predict y given z whereas the encoder is trained to make it impossible for the discriminator to predict y given z only. The decoder should reconstruct x given (z, y). At test time, the discriminator is discarded and the model can generate different versions of x when fed with different attribute values.

4 Reproduction of Papers Results

In this section we present the methods used by the authors in the paper, as well as their results and our results from reproducing their models.

C_k is a convolution-BatchNorm-ReLU layer with k filters. Convolutions use kernel of size 4×4, with a stride of 2, and padding of 1. They use leaky-ReLUs with a slope of 0.2 in the encoder, and simple ReLUs in the decoder.

The encoder is composed of the following 7 layers:

$$C_{16} - C_{32} - C_{64} - C_{128} - C_{256} - C_{512} - C_{512} \tag{4}$$

To provide the decoder with image attributes, they append the latent code to each layer given as input to the decoder, where the latent code of an image is the concatenation of the one-hot vectors representing the value of its attributes. They append the latent code as additional constant input channels for all the convolutions of the decoder. Denoting n the number of attributes, (hence a code of size $2n$), the decoder is symmetric to the encoder, but uses transposed convolutions for the up-sampling:

$$C_{512+2n} - C_{512+2n} - C_{256+2n} - C_{128+2n} - C_{64+2n} - C_{32+2n} - C_{16+2n} \tag{5}$$

The discriminator is a C_{512} layer followed by a fully-connected neural network of two layers of size 512 and n respectively.

Dropout The authors add dropout [19] to the discriminator, with a dropout rate of 0.3 in all their experiments. Following [7], the authors also tried to add dropout in the first layers of the decoder, but this turned out to significantly decrease the performance.

Discrimination cost scheduling. The authors use a variable weight for the discriminator loss coefficient λ_E. They set λ_E to 0 and the trained the model like a normal auto-encoder. λ_E is linearly increase to 0.0001 over the first 500,000 iterations to gradually make the model produce invariant representations. The authors note that this was crucial in their experiments as without it they observed the encoder was effected too much by the loss coming from the discriminator, even for low values of λ_E.

Model selection. The authors use Mean Squared Error (MSE) to measure the reconstruction error on original images, and used a classifier to predict image attributes to check if the model properly swapped the attributes of an image. At the end of each epoch, they swap the attributes of each image in the validation set and measure how well the classifier performed on the decoded images. Finally, they use human evaluation on images from the train set reconstructed with swapped attributes.

4.1 Experiment

Setup

The authors use the celebA dataset [13], which contains 200,000 images of celebrities of shape 178 $\times$ 178 annotated with 40 attributes. We limit our experiments to the modification of the "Male" attributes *only*, due to limited computational resources. For pre-processing, they crop the images to 178 $\times$ 178, and re-size them to 256 $\times$ 256, which is the resolution used in all figures of the paper. Image values are normalized to [-1, 1]. All models are trained with Adam [9], using a learning rate of 0.002, Adam's exponential decay rate for the first moment estimation variable β_1 = 0.5, and a batch size of 32. They perform data augmentation by flipping horizontally images with 0.5 probability at each iteration. As model baseline, they use IcGAN [18] with the model provided by the authors and train it on the same dataset. In our experiments, due to limited computational resources, we restricted our experiments to image sizes of 128 $\times$ 128 instead of 256$\times$256, and made our validation splits off of 120,000 training samples instead of the entire dataset. The λ_E trade-off term was scheduled to linearly increase every 300,000 iterations instead of 500,000 iterations. We found this hyperparameter to be optimal due to using different dimensions than the ones proposed in the original paper.

Quantitative evaluation method

The authors evaluation of their Fader Networks is based on two criterion: the naturalness, that measures the quality of the generated images, and the accuracy, that measures how well the attribute was swapped. Given the unsupervised nature of this task, the authors used Mechanical Turk [11] to evaluate their criterion. They produce the images using their method and IcGANs as a baseline. They compare the real image without any transformations, FadNet AE and IcGAN AE, that reconstruct original images without attribute alterations, and FadNets Swap and IcGAN Swap, that generates the images with one swapped attribute. In their study, they swapped the Mouth, Gender and Glasses attributes.

The authors use Mechanical Turk workers to evaluate their models which we cannot use due to financial constraints. As such, we implemented two neural networks that proxy the Mechanical Turk, which we call Neural Mechanical

Turk for accuracy and naturalness which we denote as NMT-A and NMT-N, respectively. The NMT's will provide us with the accuracy and naturalness metrics of the competing models.

NMT-A. The accuracy metric is a standard ResNet-18 [21] which classifies the generated images for a specified attribute on the same dataset. We trained the ResNet-18 to distinguish males from females on 20,000 celebrity images with a batch size of 64 on 10 epochs, which was enough for us to get approximately 95.0% accuracy. The training and validation ratio used was 4:1. We used the standard cross-entropy loss and normalized the image from their original integer domain [0, 255] to a real domain [0,1].

NMT-N. The modified naturalness metric that is implemented for the generated images is the Fréchet Inception Distance (FID) [6]. FID makes use of the Inception Network [20] to extract features from intermediate layers and then model those features' data distribution as a Gaussian. FID is calculated by the following equation:

$$FID(x, g) = ||\mu_x - \mu_g||^2 + Tr(\sum_x + \sum_g -2(\sum_x \sum_g)^{\frac{1}{2}}) \tag{6}$$

Where x represents the real images and g represents the generated images, and Tr is used to sum up the diagonal elements of the respective covariance matrices for x and g. FID assesses covariance matrices and means of the two inferred Gaussians, thus giving us a measure of how strongly connected two datasets are. If the FID measure is lower, this corresponds to the generated images being similar to the real images, and hence we associate this to a higher "naturalness" of the image. As Heusel et al.[6] noted, FID is found to be close to human evaluation, justifying our choice of using the metric. We did not choose the use of Inception Score (IS), a very popular metric in the literature, due to the fact that it doesn't rely on the statistics of real world samples and generated samples to formalize a statement about quality and diversity of the two datasets. Moreover, FID is more robust to high variance in the dataset than IS, as noted by Heusel et al. In all of our experiments, we use a custom implementation [16] as opposed to the original implementation on Tensorflow, due to compatibility issues.

Indeed the NMT's are proxies for the Mechanical Turk and will result in different results from the paper. On this note, we stress that with this scheme, we are not able to assess the difference between the original reported results and our results, due to using different metrics. Despite this short coming, the NMT's will still be of much use throughout our ablation study as it provides us with a rough indication of how our different models perform under various conditions, comparing to a baseline FID calculated from replicating the same experiment. Further, it will act as the baseline to which we compare any alterations to the models.

Results

Using almost the same hyperparameters as the authors, we implement the FadNet on the celebA data set by swapping genders of people in the image. We follow the same method of evaluation by using the real images as a control, FadNet AE and FadNet Swap as the competing models. Table 1 shows the results of the Fader Network paper's and ours results. Note that the Replication Naturalness is an FID score, with a value closer to zero being preferred.

Model	Naturalness		Accuracy	
	Lample et al. (%)	Replication (FID score)	Lample et al. (%)	Replication (%)
Real Image	88.6	3.8	97.6	94.7
FadNet AE	78.8	32.6	94.5	98.8
FadNet Swap	45.3	41.3	76.6	99.7

Table 1: Evaluation of naturalness and swap accuracy for each model by swapping the gender attribute. Under Lample et al. The naturalness score is the percentage of images that were labeled as "real" by human evaluators to the question "Is this image a real photograph or a fake generated by a graphics engine?", and the accuracy score is the classification accuracy by human evaluators on the values of each attribute. Under Replication, naturalness score is the FID score from the NMT-N (the lower the better), and the accuracy score is the accuracy from the NMT-A

As expected, the results from the authors paper and ours differ. Given that we used two different evaluation methods this is normal. Furthermore, we used slightly different hyperparameters as well as different image dimensions, the crucial parameter to get the results right. Nevertheless, the replication results is the best proxy results we can produce to perform robustness tests on the FadNet given our resources available.

5 Ablation

To test the robustness of the model and evaluate the importance of its various components, we examine the number of encoder layers, excluding dropout, discriminator cost scheduling and various loss functions. The baseline results that we will compare all ablation results to is the results derived above in Table 1.

We focus on the effects of swapping gender attributes in order to demonstrate the behavior of the model with respect to the changes that are conducted. Experiments are carried out in a "all else remain equal" fashion, meaning that if it hasn't been explicitly stated, one can assume we are using the original hyperparameters and implementations. This is to avoid later confusion.

5.1 Loss Functions

The loss function for the fader network in the original paper is the classical MSE loss function (1). In order to observe the changes in the quality of the images, we use several loss functions such as Mean Absolute Error (MAE), Huber Loss in place of the original MSE from the autoencoder loss, and PatchGAN [7] instead of the standard discriminator loss in the model's overall loss function (3). Table 2 displays the results under the various loss functions.

Using either the MAE loss or the Huber loss results in no significant differences or changes in the generated images compared to the author's original implementation of MSE, as justified by FID scores obtained from a generated dataset using a model trained with those losses. This result may be because of the lack of drastic outliers within the dataset. MSE harshly penalizes points which are outliers due to the squared term, resulting in a higher error, while MAE does not and is more robust to outliers. Therefore due to the lack of significant outliers, the MSE and MAE are behaving as if they were the same loss function. The Huber Loss function encompasses both the MSE and the MAE as around the origin by some hyperparameter delta, the loss function resembles that of the MSE and beyond delta, resembles L1 loss. Our hypothesis of a similar behavior to MSE and MAE was proven, as there was no drastic change in the accuracy and FID reported.

However, with the PatchGAN implementation as an adversarial loss along with MSE, we found that it decreased the sharpness and quality of the generated images. The blurry result may be caused by the distribution of the data, which is highly important. Ideally for PatchGAN, the dataset would consist of several images of the same kind of face and an equal amount of images for each kind. In the celebA dataset, the images are of different people, which results in a non-uniform distribution of the types of faces PatchGAN would be considering. Figure 2 shows the results from the various loss functions.

Model	Naturalness			Accuracy		
	MAE	Huber Loss	PatchGAN	MAE	Huber Loss	PatchGAN
FadNet AE	34.8	35.0	87.5	93.3	94.5	82.2
FadNet Swap	50.0	45.5	99.9	96.8	97.8	83.0

Table 2: Evaluation of naturalness and swap accuracy for each model by swapping the gender attribute. The Replication baseline results are compared to the results for the FadNet when implemented with Huber Loss, MAE and PatchGAN loss. Naturalness score is the accuracy from the NMT-N, and the accuracy score is the accuracy from the NMT-A

5.2 Encoder and decoder layers

Our experiments with modifying encoder and decoder layer sizes conclude that the model works reasonably well at 5 to 7 convolutional blocks both in the encoder and the decoder, which are the setups we used to generate the results reported in this paper. However, any number beyond that threshold is sure to return poor results. We hypothesize this is due to the fact that the neural network becomes highly sensitive to noise, thus a higher variance, as it picks up weaker features that do not necessarily contribute to the encoding or the decoding process (i.e. changes in the background of the image). We compare the replication results with FadNet with 4 and 8 layers, respectively. The results are shown in Table 3.

By changing the layers, we note that naturalness increases for both 4 and 8 layers, however changing the layers decreases the accuracy for both 4 and 8 layers relative to the replication results. This is similar to the findings of the authors as they noted that optimal range of layers was between 5-7, and deviating from this range would reduce performance, which it has on both naturalness and accuracy.

(a) Original Image (b) Replication,MSE (c) Huber (d) PatchGAN (e) MAE (f) SkipConnect

Figure 2: Quality of images after transformations under various loss functions with the FadNet Swap.

Model	Naturalness (NTM-N)			Accuracy		
	Replication	4 Layers	8 Layers	Replication	4 Layers	8 Layers
Real Image	3.8	-	-	94.7	-	-
FadNet AE	32.6	45.5	62.7	98.8	97.4	86.9
FadNet Swap	41.3	52.3	79.8	99.7	97.4	78.5

Table 3: Evaluation of naturalness and swap accuracy for each model by swapping the gender attribute. The Replication baseline results are compared to the results for the FadNet when implemented with 4 and 8 layers, respectively. Naturalness score is the accuracy from the NMT-N, and the accuracy score is the accuracy from the NMT-A

5.3 Dropout

We analyze the effects of excluding dropout in the discriminator. We set the dropout rate to 0.3. As the authors attempted to implement dropout in the first layers of the decoder but found it to decrease the results, we did not implement it here either.

Model	Naturalness		Accuracy	
	Replication (p = *0.3*)	No Dropout	Replication (p = *0.3*)	No Dropout
Real Image	3.8	-	94.7	-
FadNet AE	32.6	32.0	98.8	98.8
FadNet Swap	41.3	45.9	99.7	98.0

Table 4: Evaluation of naturalness and swap accuracy for each model by swapping the gender attribute. The Replication baseline results are compared to the results for the FadNet when implemented with out drop out. Naturalness score is the accuracy from the NMT-N, and the accuracy score is the accuracy from the NMT-A

We realize that despite Lample et al. claiming that discriminator dropouts were essential to the swapping quality, our experimental results proved the contrary as there is negligible difference between FID scores of dropout and no-dropout models, as well as accuracy (Table 4). Although there are strong benefits in using dropouts as a regularizer for the discriminator, we argue that penalizing the λ_E scheduling term in the overall loss function acts as a regularizer as the gradual linear increment penalizes the fader loss in the long run. Since we ran our experiments at a lower λ_E scheduling value than the authors (300,000 iterations), our model has stronger regularization as the loss gets taxed on faster.

6 Model Extensions: Skip Connections

Skip connections are extra connections between nodes in different layers of a neural network that skip one or more layers of nonlinear processing. Applying skip connections within layers has been shown to improve the overall training of a model [17]. Within our model, we implement two skip connections, one between the C_{16} layer of our encoder and C_{16} of our decoder, and the second between the C_{32} layer of our encoder and C_{32} of our decoder. We hypothesize that adding these skip connections will augment the quality of the reconstructed image, because now not only our decoding layer is getting input from the latent space but also from the original unmodified image. Table 5 shows the results from implementing Skip Connects. Despite this gain in quality, as our experiments have demonstrated, we observe a decreased capability of the model to swap attributes, reflected by the end results after swapping looking almost nearly identical to the original images. It could be that the decoder was overpowered by the skip connections as they were weighted heavier than the actual input from the latent space and the corresponding swapped attributes assigned to the images. This effect is observed as shown by Figure 2 (f) where there is an obvious superior reconstruction quality, however a poor attribute swapping quality.

Model	Naturalness		Accuracy	
	Replication	Skip Connect	Replication	Skip Connect
Real Image	3.8	-	94.7	-
FadNet AE	32.6	32.5	98.8	98.0
FadNet Swap	41.3	35.0	99.7	56.6

Table 5: Evaluation of naturalness and swap accuracy for each model by swapping the gender attribute. The baseline results are compared to the FadNet with Skip Connect. Naturalness score is the accuracy from the NMT-N, and the accuracy score is the accuracy from the NMT-A

We highlight the fact that the swapping accuracy significantly decreased. As stated before, this is evidence that skip-connections overpower connections from decoded layers. From the choice of the loss function, we hypothesize that the reconstruction loss part of the fader loss function had greater gradients during backpropagation than the discriminator loss. To avoid this problem with skip connections, as in maintaining a minimum reconstruction loss while keeping the accuracy stable, we suggest regularizing the reconstruction loss only.

7 Discussion

We reserve this section to discuss the way fader networks swap attributes and hallucinate a different version of some original image, as analyses of our results has been provided. As various failures in wrong hyperparameter tuning have shown, often times the resulting sets of images with swapped attributes resemble nothing more than the original image (with some reconstruction loss) but juxtaposed with the same template of attributes. For example, with a fader network trained to be gender invariant, pictures of various women whose feminine attribute have been swapped for masculine have the same pattern of facial hair and jawline juxtaposed onto them. We understand this generic ”male” template to be the model’s probabilistic inference of what most likely contributes to an image being labeled as ”male” versus ”female”. This template was learned during the training process of the network, and at test time, the fader network essentially tries to shape the original latent space according to that template. It could then be understood that the template is a form of neural understanding of the features, as it is actually not only manipulating local pixels but entire regions of pixels. We conjecture that this same observation could be made with other attributes as well, i.e. the network tries to put similar pairs of eyewear onto different images, for instance.

8 Conclusion

In this paper, we performed an ablation study, as well as extended the Fader Network created by Lample et al. that generates variations of images by changing the gender attribute of images on the celebA dataset. The Fader Network architecture is based on enforcing the invariance of the latent space with respect to the image attributes. The main advantage of this network is that, unlike the recent literature of image transformations that most notably use the GAN architecture, it does not apply a GAN to the decoder output. This allows it to be used in areas other than image generation such as speech, or text, where the backpropogation through the decoder can be challenging due to the non-differentiable text generation process. Our ablation study found that the results in the Lample et al. paper hold true. However, we found that the autors claims about the importance of the dropout in the discriminator to not be that important as it is regularized by the λ_E scheduling parameter.

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[2] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016.

[3] Harrison Edwards and Amos J. Storkey. Censoring representations with an adversary. *CoRR*, abs/1511.05897, 2016.

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

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

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[11] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes, 2017.

[12] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016.

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[14] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. In *Advances in neural information processing systems*, pages 981–990, 2017.

[15] Michaël Mathieu, Junbo Jake Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. Disentangling factors of variation in deep representations using adversarial training. *CoRR*, abs/1611.03383, 2016.

[16] mseitzer. pytorch-fid, February 2018.

[17] A. Emin Orhan and Xaq Pitkow. Skip connections eliminate singularities, 2017.

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

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[21] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[22] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation, 2016.

[23] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. *CoRR*, abs/1512.00570, 2015.

[24] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic photo adjustment using deep neural networks, 2014.

[25] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. Semi-latent GAN: learning to generate and modify facial images from attributes. *CoRR*, abs/1704.02166, 2017.