Conditional image generation and manipulation for user-specified content

Stap, D.; Bleeker, M.; Ibrahimi, S.; ter Hoeve, M.

DOI
10.48550/arXiv.2005.04909

Publication date
2020

Document Version
Author accepted manuscript

Conditional Image Generation and Manipulation for User-Specified Content

David Stap[*] Maurits Bleeker Sarah Ibrahimi Maartje ter Hoeve
University of Amsterdam
{d.stap,m.j.r.bleeker,s.ibrahimi,m.a.terhoeve}@uva.nl

Abstract

In recent years, Generative Adversarial Networks (GANs) have improved steadily towards generating increasingly impressive real-world images. It is useful to steer the image generation process for purposes such as content creation. This can be done by conditioning the model on additional information. However, when conditioning on additional information, there still exists a large set of images that agree with a particular conditioning. This makes it unlikely that the generated image is exactly as envisioned by a user, which is problematic for practical content creation scenarios such as generating facial composites or stock photos. To solve this problem, we propose a single pipeline for text-to-image generation and manipulation. In the first part of our pipeline we introduce textStyleGAN, a model that is conditioned on text. In the second part of our pipeline we make use of the pre-trained weights of textStyleGAN to perform semantic facial image manipulation. The approach works by finding semantic directions in latent space. We show that this method can be used to manipulate facial images for a wide range of attributes. Finally, we introduce the CelebTD-HQ dataset, an extension to CelebA-HQ, consisting of faces and corresponding textual descriptions.

1. Introduction

Conditional image generation has experienced rapid progress over the last few years [18, 19, 25, 23]. In this task, an image is generated by some sort of generative model which is conditioned on a number of attributes or on a textual description. For content creation scenarios such as generating CGI for animation movies, making forensic sketches of suspects for the police, or generating stock photos, it would be beneficial to have the chance to modify the image after initial generation to close the gap between the user requirements and the output of the model. This is important because a large number of images can adhere to a given description. The final image can then be generated by a back and forward consultation of the individual who has the "ground truth" of the description in mind.

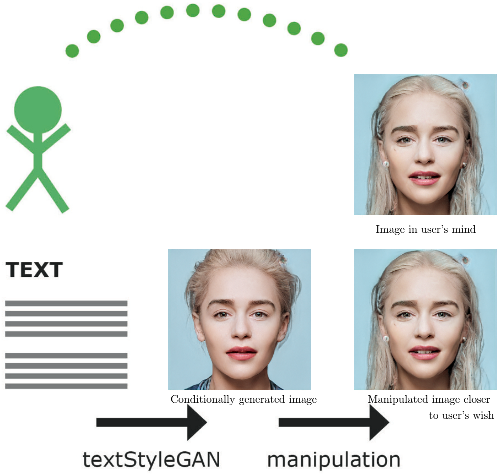


Image in user's mind
TEXT
Conditionally generated image
Manipulated image closer to user's wish
textStyleGAN
manipulation

Figure 1. **Generating a user-specified image.** A first approximation is generated using a textual description. The resulting image is then further manipulated such that it is closer to the user's desire.

In this work we focus on the generation of these user-specified images. The overarching question we aim to answer is: *How can we generate and manipulate an image, based on a fine-grained description, such that it represents the image a person had in mind?* In particular, we focus on the facial domain and aim to generate and manipulate facial images based on fine-grained descriptions. Since there exists no dataset of facial images with natural language descriptions, we create the *CelebFaces Textual Description High Quality (CelebTD-HQ)* dataset. The dataset consists of facial images and fine-grained textual descriptions.

Figure 1 shows the pipeline of our method: first we generate an initial version of an image, based on a description. Then we modify this generated image to be able to better meet the image a person had in mind. To the best of our knowledge we are the first to combine the generation and manipulation in a single pipeline. The manipulation step is essential for practical content creation applications that aim

[*] This paper is the product of an internship at the Dutch National Police

to generate the image a person had in mind.
With this work, we contribute

(C1) A text-to-image model, *textStyleGAN*, which can generate images at higher resolution than other text-to-image models and beats the current state of the art in the majority of cases;

(C2) A method for semantic manipulation that uses textStyleGAN weights. We show that these conditional weights can be used for semantic facial manipulation;

(C3) An extension to CelebA-HQ [7] dataset, where we use the attributes to generate natural sounding textual descriptions. We call this new dataset the *CelebFaces Textual Description High Quality (CelebTD-HQ)*. We share the dataset to facilitate future research.[1]

2. Related work

Our work is related to work on text-to-image synthesis (first part of our pipeline) and semantic image manipulation (second part of our pipeline). In this section we describe relevant work in both areas. We also discuss StyleGAN [8], which plays an important role in our architecture.

Text-to-image synthesis The goal of text-to-image synthesis is to generate an image, given a textual description. The image should be visually realistic and semantically consistent with the description. Most text-to image-synthesis methods are based on Generative Adversarial Networks (GANs) [5]. Pioneering work by Reed *et al*. [19, 18] shows that plausible low resolution images can be generated from a textual representation. Zhang *et al*. [25, 26] propose StackGAN, which decomposes the problem into several steps; a Stage-I GAN generates a low resolution image conditioned on text, and higher Stage GANs generate higher resolution images conditioned on the results of earlier stages and the text. More recently, Xu *et al*. [23] proposed AttnGAN, which exploits an attention mechanism to focus on different words when generating different image regions. However, guaranteeing semantic consistency between the text description and generated image remains challenging. As a solution, Qiao *et al*. [17] introduce a semantic text regeneration and alignment (STREAM) module, which regenerates the text description for the generated image. Yin *et al*. [24] propose a Siamese mechanism in the discriminator, which distills the *semantic commons* while retaining the *semantic diversities* between different text descriptions for an image. Note that all models discussed in this section do not have a manipulation mechanism, which renders them impractical for content creation.

Semantic image manipulation A relatively complex type of image modification is *semantic* manipulation, which can be thought of as changing attributes such as pose, expression or gender. The manipulation should preserve the realism and unedited factors should be left unchanged. Perarnau *et al*. [15] propose an extension to conditional GANs [13], where an image is encoded to obtain a vector image representation and a conditional binary representation. This binary representation refers to facial characteristics such as gender, hair colour and hair type. Editing is done by changing the binary representation. Shen *et al*. [21] propose a framework consisting of two generators which perform inverse attribute manipulation given an input image, *e.g*. adding glasses and removing glasses. Chen *et al*. [3] propose an encoder-decoder architecture, that models face effects with middle-level convolutional layers. In later work, Chen *et al*. [4] introduce a *semantic component model*, that does not rely on a generative model. A to be edited attribute is decomposed into multiple semantic components, each corresponding to a specific face region, which are then manipulated independently. A method based on a Variational Autoencoder [9] is introduced by Qian *et al*. [16]. Better disentanglement, where the goal is to separate out (disentangle) features into disjoint parts of a representation, is obtained by clustering in latent space. An image and target facial structural information are both encoded to latent space, and an output image is generated based on the image appearance and target facial structural information.

StyleGan Karras *et al*. [8] introduce StyleGAN to generate images – an *unconditional* generative model. StyleGAN uses a mapping network: instead of feeding the generator a noise vector directly, first a non-linear mapping is applied. Karras *et al*. show empirically that this mapping better disentangles the latent factors of variation. This implies that attributes, such as gender, are easily separable in the resulting latent space allowing for easier semantic manipulation. Furthermore, StyleGAN supports higher resolutions up to 1024×1024.

Our current work differs from the work discussed in this section in the following important ways: First, our model combines text-to-image generation and semantic manipulation, which is important for content creation. Second, we introduce a conditional variant of StyleGAN instead of building on AttnGAN [17, 24]. This enables us to generate and manipulate at higher resolutions, allowing for more fine-grained details. Third, in contrast to earlier work on semantic manipulation, *e.g*. [21, 3, 4, 16], we make use of the latent space of a trained GAN to find semantic directions. Our method is simple and computationally cheap, since it only requires a classification step, without a need to retrain the GAN or extra computationally expensive modules such as an additional Generator.

[1] https://github.com/davidstap/CelebTD-HQ

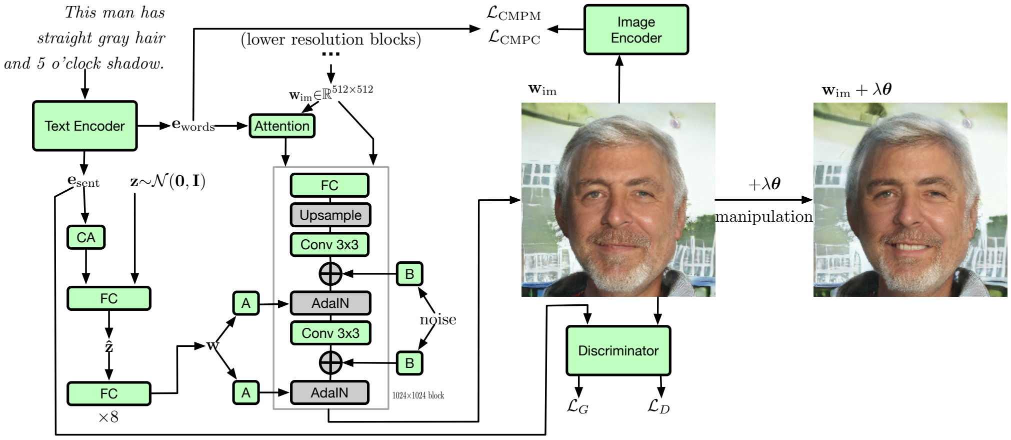

This man has straight gray hair and 5 o'clock shadow.
Text Encoder
$\mathbf{e}_{\text{words}}$
Attention
(lower resolution blocks)
$\mathbf{w}_{\text{im}} \in \mathbb{R}^{512\times512}$
$\mathcal{L}_{\text{CMPM}}$
$\mathcal{L}_{\text{CMPC}}$
Image Encoder
$\mathbf{w}_{\text{im}}$
$\mathbf{w}_{\text{im}} + \lambda\boldsymbol{\theta}$
$+\lambda\boldsymbol{\theta}$
manipulation
$\mathbf{e}_{\text{sent}}$
$\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
CA
FC
$\hat{\mathbf{z}}$
FC
×8
$\mathbf{w}$
A
A
FC
Upsample
Conv 3x3
AdaIN
Conv 3x3
AdaIN
B
B
noise
1024×1024 block
Discriminator
$\mathcal{L}_G$
$\mathcal{L}_D$

Figure 2. **Overview of our approach.** An image is generated from a textual description (Section 3.1). The images are generated progressively, starting from a low resolution. We only depict a single Generator block for clarity. The attention model retrieves the most relevant word vectors for generating different sub-regions. The $\mathcal{L}_{\text{CMPM}}$ and $\mathcal{L}_{\text{CMPc}}$ losses measure semantic consistency between the text and the generated image. The resulting image can be manipulated by making use of semantic directions in latent space (Section 3.2).

3. Method

In this section we describe our approach, in which we combine text-to-image synthesis (Part 1) and semantic manipulation (Part 2), in full detail. An overview is given in Figure 2. We conclude this section by a description of our new CelebTD-HQ dataset.

3.1. Part 1 - textStyleGAN

The first part of our pipeline consists of a text-to-image step. This step is visually depicted in the left part of Figure 2. We condition StyleGAN [8] on a textual description.

Generator For the generator, we combine latent variable $\mathbf{z} \in \mathbb{R}^{D_z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and text embedding $\mathbf{e}_{\text{sent}} \in \mathbb{R}^{512}$ by concatenation, and perform a linear mapping, resulting in $\hat{\mathbf{z}} = \mathbf{W}[\mathbf{z}; \mathbf{e}_{\text{sent}}]$ with $\mathbf{W} \in \mathbb{R}^{D_z \times D_z + 512}$.

Representing textual descriptions In order to obtain text representation $\mathbf{e}_{\text{sent}}$, we make use of a recent image-text matching work [27].[2] First we pre-train a bi-directional LSTM (for text) and a CNN (for images) encoder to learn a joint embedding space for text and images using two loss functions: 1) the cross-modal projection matching (CMPM) loss, which minimizes the KL divergence between the projection compatibility distributions and the normalized matching distributions and 2) the cross-modal projection classification (CMPC) loss, making use of an auxiliary classification task. We then add these pre-trained encoders to textStyleGAN, use the text encoder to obtain sentence feature $\mathbf{e}_{\text{sent}}$ and word features $\mathbf{e}_{\text{words}}$, and the image encoder to encode generated images.[3]

Conditioning Augmentation Instead of directly using the textual representation $\mathbf{e}_{\text{sent}}$ for image generation, we use Conditioning Augmentation (CA) [25]. This encourages smoothness in the latent conditioning manifold. We sample from $\mathcal{N}(\mu(\mathbf{e}_{\text{sent}}), \Sigma(\mathbf{e}_{\text{sent}}))$ where the mean $\mu(\mathbf{e}_{\text{sent}})$ and covariance matrix $\Sigma(\mathbf{e}_{\text{sent}})$ are functions of embedding $\mathbf{e}_{\text{sent}}$. These functions are learned using the reparameterization trick [9]. To enforce smoothness and prevent overfitting we add the following KL divergence regularization term during training:

$$D_{\text{KL}}\Big(\mathcal{N}(\mu(\mathbf{e}_{\text{sent}}), \Sigma(\mathbf{e}_{\text{sent}}))||\mathcal{N}(\mathbf{0}, \mathbf{I})\Big). \tag{1}$$

We then feed $\hat{\mathbf{z}}$ through eight fully connected layers to create the more disentangled representation $\mathbf{w}$, according to the original StyleGAN generator architecture. From this $\mathbf{w}$ an image is then generated.

Attentional guidance Instead of only using the final sentence representation $\mathbf{e}_{\text{sent}} \in \mathbb{R}^D$, the Generator also makes use of the intermediate representations $\mathbf{e}_{\text{words}} \in \mathbb{R}^{D\times T}$ for attentional guidance [23]. Specifically, the attentional guidance module takes as input word features $\mathbf{e}_{\text{words}}$ and image features $\mathbf{h}$. The word features are first converted to the same dimensionality by multiplying with a (learnable) matrix, i.e. $\mathbf{e}'_{\text{words}} = \mathbf{U}\mathbf{e}_{\text{words}}$. Then, a word-context vector $\mathbf{c}_j$ is computed for each subregion of the image based on its hidden features $\mathbf{h}$ and word features $\mathbf{e}_{\text{words}}$. Each column of $\mathbf{h}$ is a feature vector of a sub-region of the image. For the i^{th}

[2] Recent text-to-image works [23, 17, 24] use a similar but inferior type of visual-semantic embedding that was introduced by Reed *et al.* [18]. For a fair comparison we have experimented with these visual-semantic embeddings, but did not find any significant difference. We use the embeddings by [27], as we will make use of the corresponding image decoder for cross-modal similarity enhancement.

[3] We experimented with fine-tuning the text and image encoder but did not experience improved performance.

subregion, $\mathbf{c}_i$ is a dynamic representation of word vectors relevant to $\mathbf{h}_i$, which is calculated by

$$\mathbf{c}_i = \sum_{j=1}^{T} \boldsymbol{\alpha}_{ij} \mathbf{e}'_j. \tag{2}$$

The matrix $\boldsymbol{\alpha}_{ij}$—which indicates the weight the model attends to the j^{th} word when generating the i^{th} subregion of the image—is computed by

$$\boldsymbol{\alpha}_{ij} = \frac{\exp\Big(\text{score}(\mathbf{h}_i, \mathbf{e}_j)\Big)}{\sum_{k=1}^{T} \exp\Big(\text{score}(\mathbf{h}_i, \mathbf{e}_k)\Big)}, \tag{3}$$

where

$$\text{score}(\mathbf{h}_i, \mathbf{e}_j) = \mathbf{h}_i^{\mathrm{T}} \mathbf{e}'_j \tag{4}$$

is the dot score function [11] which can be thought of as similarity score between the image sub-region and word. The Generator receives attentional guidance for resolutions of 64×64 and upwards.

Discriminator As for the discriminator, we feed it with an image (either real or fake) and corresponding text embedding $\mathbf{e}_{\text{sent}}$. The image is fed through nine convolutional layers (one for each intermediate resolution in the case of 1024×1024 images, according to the original StyleGAN discriminator architecture) and a fully connected layer, resulting in feature representation $\mathbf{h} \in \mathbb{R}^{D_h}$. We then append $\mathbf{e}_{\text{sent}}$ and feed the resulting vector through a final fully connected layer, where the Discriminator outputs a value that represents the probability of the image being real, resulting in loss signals $\mathcal{L}_G$ and $\mathcal{L}_D$.

Training loss Following common practice in conditional synthesis [25, 26, 23] we employ two adversarial generator losses: an unconditional loss, measuring visual realism and a conditional loss that measures semantic consistency.

$$\mathcal{L}_G = \underbrace{-\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_G}[\log D(\mathbf{x})]}_{\text{unconditional loss}} \underbrace{-\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_G}[\log D(\mathbf{x}, \mathbf{e}_{\text{sent}})]}_{\text{conditional loss}}. \tag{5}$$

The discriminator is trained to classify the input image $\mathbf{x}$ as real or fake by minimizing the loss

$$\begin{aligned}\mathcal{L}_D = &\underbrace{-\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x})] - \frac{1}{2}\mathbb{E}_{\mathbf{x}}\sim p_G[\log 1 - D(\mathbf{x})]}_{\text{unconditional loss}} \\ &\underbrace{-\frac{1}{2}\mathbb{E}_{\mathbf{x}}\sim p_{data}[\log D(\mathbf{x}, \mathbf{e}_{\text{sent}})] - \frac{1}{2}\mathbb{E}_{\mathbf{x}}\sim p_G[\log 1 - D(\mathbf{x}, \mathbf{e}_{\text{sent}})]}_{\text{conditional loss}}.\end{aligned} \tag{6}$$

To additionally ensure semantic consistency between textual descriptions and generated images, we make use of cross-modal similarity loss, *i.e.* CMPM and CMPC losses [27] $\mathcal{L}_{\text{CMPM}}$ and $\mathcal{L}_{\text{CMPC}}$. We use word representations $\mathbf{e}_{\text{words}}$, and encode the generated image to calculate these losses.

3.2. Part 2 - Semantic manipulation by latent sample classification

The conditionally generated images from Part 1 are unlikely to correspond exactly to what a user had in mind when giving a certain description. To account for this we extend our model such that a generated image can be manipulated, by making use of semantic directions in latent space. Although this method is data agnostic, we highlight the facial domain. Using this method for other domains is highly similar.

In the remainder of this section we describe our method for finding a wide range of directions, which we call *latent sample classification*. We make use of the disentangled textStyleGAN latent space. The first key insight of the second part of our method is that we make use of the latent textStyleGAN space $\mathcal{W}$ instead of $\mathcal{Z}$. Using the latter would require us to solve the more challenging task of recovering the conditioning of an image when finding its latent representation.

Our method can be described in four steps:

1. Sample n images from a pre-trained (text)StyleGAN generator.
2. Perform classification for images on a single attribute (*e.g.*, smiling / not smiling, male / female) to obtain labels.
3. Train a logistic regression classifier on a single attribute to find direction.
4. Resulting weight $\boldsymbol{\theta}$ is equivalent to the desired direction in latent space.

We describe all four steps in more detail below.

- **Step 1** We randomly sample 1000 $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and transform these into $\mathbf{w}$, which we treat as features in subsequent steps. We established empirically by varying n that more samples do not improve the metric scores.

- **Step 2** For classification of our generated images we make use of Face API[4], which we treat as a black box classification algorithm. This service allows us to obtain classifications for `smile`, `gender` and `age`. We choose to use this service as opposed to training our own classification algorithm because we do not want to rely on labels; our aim is to devise a method that works in both the conditioned and the unconditioned case.

- **Step 3** Why do we train a linear classifier? A major benefit of intermediate latent space $\mathcal{W}$ is that it does not have to support sampling according to any fixed distribution. Instead, its sampling density is a result of the learned piecewise continuous mapping $f(\mathbf{z})$. Karras *et al.* [8] show empirically that this mapping unwarps $\mathcal{W}$ so that the factors of variation become more linear. As a result, manipulating images in this space is expected to be more flexible to a training set with arbitrary attribute distribution, i.e. manipulating a single feature is less likely to influence other highly correlated features.

  Our aim is to find the correct binary labels $\mathbf{y}$ (*e.g.*, smiling / not smiling) given features $\mathbf{w}$ (sampled latent image codes), and therefore we optimize the cross-entropy loss. This allows us to find an approximation of the optimal weights $\boldsymbol{\theta}$.

- **Step 4** Intuitively, $\mathbf{w}_{\text{im}}$ is a point in latent space $\mathcal{W}$ corresponding to an image after the transformation by the generator. The resulting weights $\boldsymbol{\theta}$ from the previous step are representing the desired (approximately linear) direction in latent space $\mathcal{W}$. By combining $\mathbf{w}_{\text{im}}$ and $\boldsymbol{\theta}$, we can take a step in the desired manipulation direction, resulting in $\mathbf{w} \in \mathcal{W}$ which (after being fed to the generator) corresponds to a manipulation of the original image $\mathbf{w}_{\text{im}}$. That is, to edit an image $\mathbf{w}_{\text{im}}$, we simply add (or subtract) $\boldsymbol{\theta}$:

$$\mathbf{w} = \mathbf{w}_{\text{im}} \pm \boldsymbol{\theta}. \tag{7}$$

  It is optional to use larger or smaller increments to for example add more or less smile, i.e. $\mathbf{w} = \mathbf{w}_{\text{im}} + \lambda\boldsymbol{\theta}$ is a valid operation if one wants more control over the outcome, which we demonstrate in Section 5.

3.3. CelebTD-HQ

We introduce the CelebTD-HQ dataset, as a first step towards creating photo-realistic images of faces given a description by a user. We build on the CelebA-HQ [7] dataset, which consists of facial images and their attributes. We use a probabilistic context-free grammar (PCFG) to generate a wide variety of textual descriptions, based on the given attributes. See Appendix for details. Figure 3 gives some examples of sentences that we generate. Following the format of the popular CUB dataset [22], we create ten unique single sentence descriptions per image to obtain more training data. We call our dataset *CelebFaces Textual Description High Quality* (CelebTD-HQ). The total dataset consists of 30000 images, which we randomly divide into 80% train and 20% test samples. In Section 4 we describe all other datasets that we use for our experiments.

4. Experimental setup

4.1. Datasets

To evaluate our pipeline we use the CUB [22] and MS COCO [10] datasets for the text-to-image part and the FFHQ [8] dataset. We use our new CelebTD-HQ dataset to show that our method for semantic manipulation works. Since our method is domain-agnostic, one could also use it for semantic manipulation with other images. We describe the first three datasets in more detail here – our CelebTD-HQ was described in Section 3.3.

- **CUB** [22] contains 11788 images of 200 bird species. Reed *et al.* [18] collected ten single-sentence visual descriptions per image. The data is divided into train and test sets according to [23], resulting in 8855 train and 2933 test samples.

- **MS COCO** [10] is comprised of complex everyday scenes containing 91 object types in their natural context. It has five textual descriptions per image. The data is divided into train and test sets according to [23], resulting in 80000 train and 40000 test samples.

- **FFHQ** [8] consists of 70000 high-quality images at 1024 × 1024 resolution. In contrast to the other datasets, it is unlabeled. This dataset is used to train StyleGAN [8]. We fine-tune pre-trained StyleGAN weights for conditional facial synthesis.

4.2. Ablation study

We present several ablations for the text-to-image task, resulting in the following architectures:

- **textStyleGAN base** The textStyleGAN base model corresponds to our architecture as presented in Section 3.1, but without Conditioning Augmentation (CA), attentive guidance and the cross-modal similarity loss;

- **with Conditioning Augmentation** This model corresponds to the textStyleGAN base model with CA;

- **with Attention** This model corresponds to the textStyleGAN base model with CA and attentive guidance;

- **with Cross-modal similarity loss** This is our complete model, *i.e.* the textStyleGAN base model with CA, attentive guidance and cross-modal similarity loss.

[4] https://azure.microsoft.com/services/cognitive-services/face/

4.3. Implementation details

Part 1 - Conditional Synthesis. TextStyleGAN builds on the original StyleGAN [8] TensorFlow [1] implementation[5]. We adopt all training procedures and hyperparameters from Karras *et al*. [8]. Training of our conditional image generation models is performed on 4 NVIDIA GeForce 1080Ti GPUs with 11GB of RAM. Although the vanilla textStyleGAN model described above fits on a single GPU, we need to decrease the model size before adding enhancements as described in the previous section. We use mixed precision training [12] and observe no performance drop.

Part 2 - Image manipulation models. For our semantic image manipulation method, we make use of the *conditional* textStyleGAN weights. Training of our image manipulation models is performed on a single NVIDIA Titan V GPU with 12GB of RAM.

4.4. Evaluation

Following common practice in text-to-image work [23, 17, 24, 29], we report Inception Score (IS) [20] by randomly sampling 30000 unseen descriptions from the test sets and R-precision (as described in [23]) on the CUB and COCO datasets. To compare facial image quality with unconditional StyleGAN, we report Fréchet Inception Distance (FID) [6] scores. Furthermore, to compare the level of disentanglement between StyleGAN and our textStyleGAN, we report perceptual path lengths [8]. To measure semantic consistency for textStyleGAN trained on CelebTD-HQ, we report classification scores of several attributes on the generated images. We evaluate semantic facial image manipulation qualitatively.

5. Results

To be able to compare to previous work we evaluate both parts of our pipeline separately on their respective tasks. We present the results in this section.

5.1. Part 1 - textStyleGAN

Following the evaluation protocol for earlier work on text-to-image for CUB and COCO [25, 26, 28, 23, 17, 24] we have calculated IS to quantify image quality. Scores are presented in Table 2. Our full text-to-image model performs best on CUB and second best on COCO, with scores of 4.78 and 33.00 respectively. Furthermore, following StyleGAN evaluation [8], FID scores for CelebTD-HQ are listed in Table 1. The results indicate that FID scores for our conditional model are slightly better than the (unconditional) StyleGAN model.

Model	CelebTD-HQ	Resolution
StyleGAN (unconditioned) [8]	5.17 ± 0.08	1024 × 1024
textStyleGAN base (ours)	5.11 ± 0.09	1024 × 1024
w/ CA	5.11 ± 0.06	1024 × 1024
w/ Attention	5.10 ± 0.06	1024 × 1024
w/ Cross-modal similarity	**5.08** ± 0.07	1024 × 1024

Table 1. Fréchet Inception Distance for our text-to-image models on CelebTD-HQ.

Model	CUB	COCO	Resolution
GAN-INT-CLS [19]	2.88 ± 0.04	7.88 ± 0.07	64 × 64
GAWWN [18]	3.62 ± 0.07	-	128 × 128
StackGAN [25]	3.70 ± 0.04	8.45 ± 0.03	256 × 256
StackGAN++ [26]	3.82 ± 0.06	-	256 × 256
PPGN [14]	-	9.58 ± 0.21	227 × 227
HDGAN [28]	4.15 ± 0.05	11.86 ± 0.18	256 × 256
AttnGAN [23]	4.36 ± 0.03	25.89 ± 0.47	256 × 256
MirrorGAN [17]	4.56 ± 0.05	26.47 ± 0.41	256 × 256
SD-GAN [24]	4.67 ± 0.09	**35.69** ± **0.50**	256 × 256
DM-GAN [29]	4.75 ± 0.07	30.49 ± 0.57	256 × 256
textStyleGAN base (ours)	3.89 ± 0.04	14.85 ± 0.50	256 × 256
w/ CA	4.01 ± 0.07	16.26 ± 0.43	256 × 256
w/ Attention	4.72 ± 0.08	32.34 ± 0.49	256 × 256
w/ Cross-modal similarity	**4.78** ± **0.03**	33.00 ± 0.31	256 × 256

Table 2. IS for various text-to-image models on CUB and COCO datasets.

In order to compare semantic consistency to earlier work, we calculate R-precision scores for CUB and COCO, presented in Table 3. The results demonstrate that the generated images are semantically consistent to their textual descriptions. Notably, the scores improve significantly with the cross-modal similarity loss. For $k = 1$, CUB scores improve from 51.52 to 74.72 and COCO scores from 73.02 to 87.02. This makes sense, since this loss explicitly forces semantic consistency between images and text. Without this loss, the model learns semantic consistency implicitly, but only to a certain extent.

Dataset	CUB			COCO		
top-k	k=1	k=2	k=3	k=1	k=2	k=3
AttnGAN [23]	53.31	54.11	54.36	72.13	73.21	76.53
MirrorGAN [17]	57.67	58.52	60.42	74.52	76.87	80.21
DM-GAN [29]	72.31	–	–	**88.56**	–	–
textStyleGAN base (ours)	40.98	42.43	45.59	60.03	61.47	63.08
w/ CA	45.06	46.50	48.33	65.84	67.88	71.59
w/ Attention	51.52	53.89	55.24	73.02	75.74	78.38
w/ Cross-modal similarity	**74.72**	**76.08**	**79.56**	87.02	**87.54**	**88.23**

Table 3. R-precision scores for various text-to-image models on CUB and COCO.

To determine semantic consistency for CelebTD-HQ, we take a different approach. We again use Face API to determine if attributes that are present in the textual description are also present in the generated image. The results in Table 4 demonstrate that this is indeed mostly the case.

We are interested in the degree of disentanglement of the latent space $\mathcal{W}$ in the case of textStyleGAN trained on CelebTD-HQ. A high degree of disentanglement is desirable because this will make manipulating the generated images easier. Therefore, we calculated perceptual path length, and compare scores to (unconditional) StyleGAN.

[5] https://github.com/nvlabs/stylegan

The scores are presented in Table 5, and indicate a similar level of disentanglement in both latent spaces. Our conditional model has a score of 201.1, slightly higher than the 200.5 score for StyleGAN.

Finally, see Figures 3, 4 and 5 for qualitative samples of our full textStyleGAN model on CUB, COCO and CelebTD-HQ.

Attribute	textStyleGAN	CelebA-HQ
Bald	0.98	1.00
Black_Hair	0.93	0.97
Blond_Hair	1.00	1.00
Brown_Hair	0.96	0.98
Eyeglasses	0.92	1.00
Gray_Hair	0.89	0.99
Makeup	0.76	0.86
Male	1.00	1.00
No_Beard	1.00	1.00
Smiling	0.87	0.93
Young	0.86	0.90

Table 4. Attribute classification scores for images generated with textStyleGAN and for the original CelebA-HQ dataset.

Model	Perceptual path length
StyleGAN (unconditional) [8]	200.5
textStyleGAN base (ours)	201.4
w/ CA	200.1
w/ Attention	200.8
w/ Cross-modal similarity	201.1

Table 5. Perceptual path length scores for StyleGAN trained on FFHQ and textStyleGAN trained on CelebTD-HQ.

5.2. Part 2 - Semantic manipulation

We present qualitative examples for attribute directions in Figures 7 (smile direction), 8 (gender direction) and 9 (age direction). The results demonstrate that our method can change single attributes while holding most others constant. However, because of coupled attributes in the biased training data, in some cases this is not possible. This can be observed in Figure 8. In the top and bottom row the subject wears earrings (which were not present before manipulation) when edited to become more female. Another interesting observation is that the color of the jacket changes from blue to red in the bottom row in Figure 7.

Removing artifacts We show that our method can improve the visual quality of images generated by StyleGAN, which suffers from circular artifacts. We use our method to find a circular artifact direction in latent space, by manually classifying 250 images into artifact or no artifact. This direction can then be subtracted from images with artifacts, often resulting in an image without artifact, as depicted in Figure 6. Most similar is work by Bau *et al.* [2], who identified units (defined as layers in the generator network) re-

(a) *The woman has her mouth slightly open and has black hair. She is smiling and young.*

(b) *This person has 5 o'clock shadow, wavy hair and bushy eyebrows. He has a big nose.*

(c) *This young man has black hair and no beard. He is smiling.*

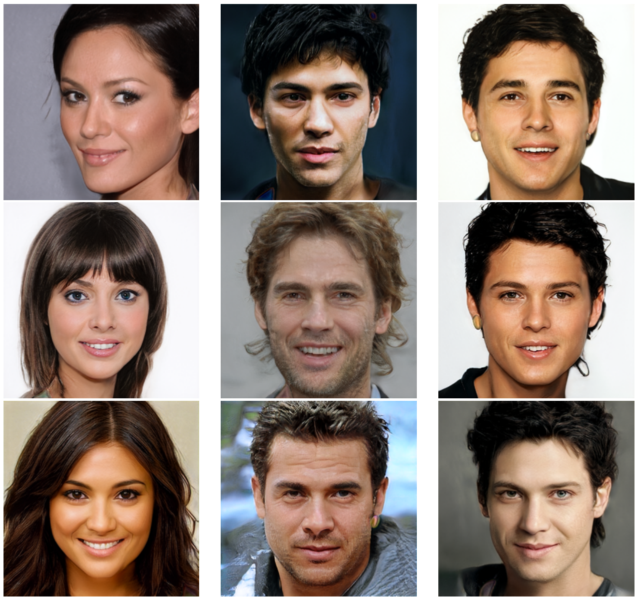


Figure 3. textStyleGAN trained on CelebTD-HQ. Different noise vectors for all images.

(a) *this bird has wings that are black with yellow belly*

(b) *this bird has a white breast and crown with red feet and grey wings.*

(c) *a small, light red bird, with black primaries, throat, and eyebrows, with a short bill.*

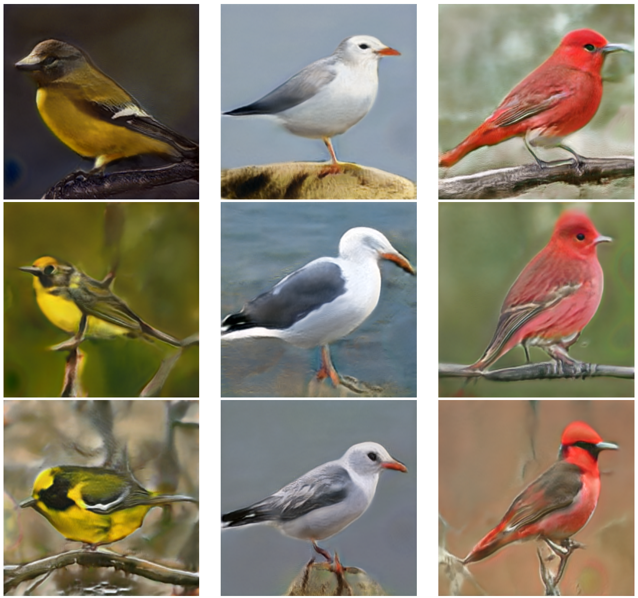
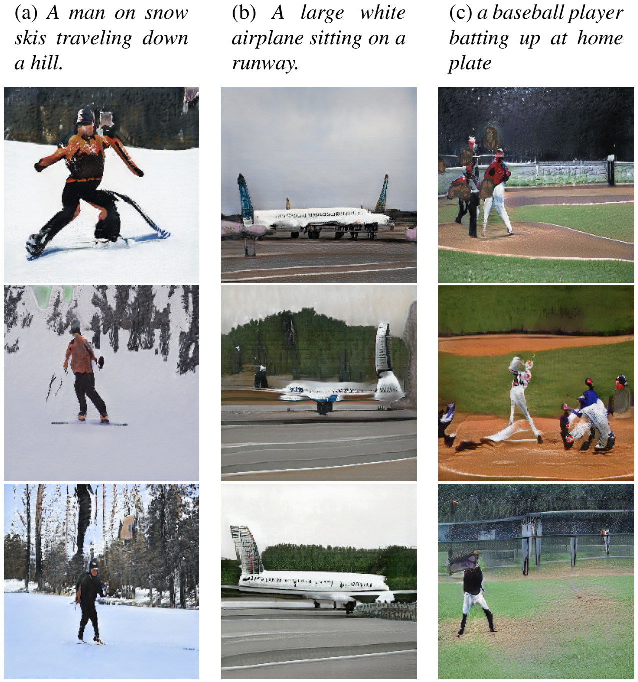


Figure 4. textStyleGAN trained on CUB. Different noise vectors for all images.


(a) *A man on snow skis traveling down a hill.*

(b) *A large white airplane sitting on a runway.*

(c) *a baseball player batting up at home plate*

Figure 5. textStyleGAN trained on COCO. Different noise vectors for all images.

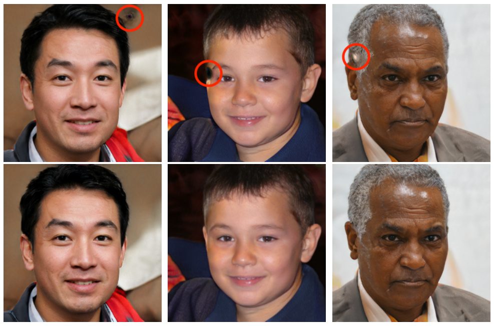


Figure 6. Our sample classification method can also be used to remove circular artifacts. The top images are sampled from StyleGAN and suffer from circular artifacts, highlighted by a red circle. These artifacts can often be removed by subtracting the circular artifact direction.

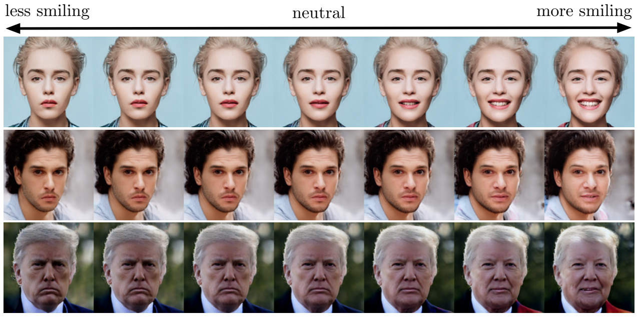


Figure 7. Qualitative results for sample classification method. From left to right: less joy, more joy.

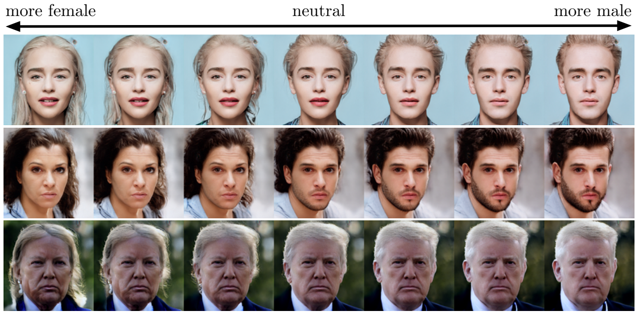


Figure 8. Qualitative results for sample classification method. From left to right: more female, more male.

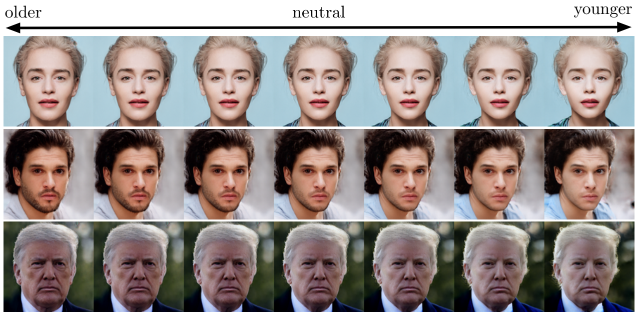


Figure 9. Qualitative results for sample classification method. From left to right: older, younger.

sponsible for artifacts, which are then ablated to suppress the artifacts. To the best of our knowledge, there is no earlier work on removing GAN artifacts by making use of directions in the latent space. StyleGAN artifacts are obvious indications of the artificial origin of an image, and being able to remove these leads to more convincing images.

6. Conclusion

This paper shows that conditional image synthesis and semantic manipulation can be brought together and utilized for practical content creation applications. We have introduced a novel text-to-image model, textStyleGAN, that allows for semantic facial manipulation after generating an image. This facilitates manipulating conditionally generated images, hence effectively unifying conditional generation and semantic manipulation. We quantitatively showed that the model is comparable to the state-of-the-art while allowing for higher resolutions. Finally, we have introduced CelebTD-HQ, a facial dataset with full length text descriptions based on attributes. We show that this dataset can be used to generate and manipulate facial images, and we believe applications such as facial stock photo creation is possible with our approach. For future work the exploration of more complex attributes for semantic manipulation can be considered.

7. Acknowledgements

This research was supported by the Nationale Politie. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

[1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and others. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*, 1(2), 2015. 6

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. *arXiv preprint arXiv:1811.10597*, 2018. 7

[3] Ying-Cong Chen, Huaijia Lin, Michelle Shu, Ruiyu Li, Xin Tao, Xiaoyong Shen, Yangang Ye, and Jiaya Jia. Facelet-bank for fast portrait manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3541–3549, 2018. 2

[4] Ying-Cong Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I Pao, Jiaya Jia, and others. Semantic Component Decomposition for Face Attribute Manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9859–9867, 2019. 2

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[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. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 6

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2018. 2, 5

[8] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprint arXiv:1812.04948*, 2018. 2, 3, 5, 6, 7

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[11] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 4

[12] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and others. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017. 6

[13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[14] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017. 6

[15] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 2

[16] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a Face: Towards Arbitrary High Fidelity Face Manipulation. *arXiv preprint arXiv:1908.07191*, 2019. 2

[17] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. MirrorGAN: Learning Text-to-image Generation by Redescription. *arXiv preprint arXiv:1903.05854*, 2019. 2, 3, 6

[18] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. 1, 2, 3, 5, 6

[19] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 1, 2, 6

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 6

[21] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4030–4038, 2017. 2

[22] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[23] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 1, 2, 3, 4, 5, 6

[24] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics Disentangling for Text-to-Image Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2327–2336, 2019. 2, 3, 6

[25] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. 1, 2, 3, 4, 6

[26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018. 2, 4, 6

[27] Ying Zhang and Huchuan Lu. Deep Cross-Modal Projection Learning for Image-Text Matching. In *Proceedings of the*

European Conference on Computer Vision (ECCV), pages 686–701, 2018. 3, 4

[28] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

[29] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5802–5810, 2019. 6

Appendix A. CelebTD-HQ dataset

Rule			Probability
S	→	NP VP	1
NP	→	Det Gender	0.5
NP	→	PN	0.5
VP	→	Wearing PN Are PN HaveWith	0.166
VP	→	Wearing PN HaveWith PN Are	0.166
VP	→	Are PN Wearing PN HaveWith	0.166
VP	→	Are PN HaveWith PN Wearing	0.166
VP	→	HaveWith PN Wearing PN Are	0.166
VP	→	HaveWith PN Are PN Wearing	0.166
Wearing	→	WearVerb WearAttributes	1
Are	→	IsVerb IsAttributes	1
HaveWith	→	HaveVerb HaveAttributes	1
Det	→	*a*	0.5
Det	→	*this*	0.5
Gender	→	**gender**	0.75
Gender	→	*person*	0.25
PN	→	**pn**	1
WearVerb	→	*wears*	0.5
WearVerb	→	*is wearing*	0.5
WearAttributes	→	**wears**	1
IsVerb	→	*is*	1
IsAttributes	→	**is**	1
HaveVerb	→	*has*	1
HaveAtttributes	→	**HaveWith**	1

Figure 10. PCFG used to generate captions for our CelebTD-HQ dataset.

See Figure 10 for the PCFG used to generate captions for our CelebTD-HQ dataset. Note that some terminal symbols have bold values, which can be thought of as an attribute list where either a single option is picked (e.g. *male* / *female* / *man* / *woman* in the case of Gender) or a list of attributes in the case of WearAttributes (e.g., glasses, hat), IsAttributes (e.g. smiling) and HaveAttributes (e.g. blonde hair). To promote diversity of the generated sentences, we choose equal probabilities when multiple options are available. (Except for the case of gender, where we mention the gender explicitly more often than not.) We only consider the active binary attributes, and ignore the inactive ones. A total of n attributes per description is randomly selected, where $n \sim \mathcal{N}(5, 1)$ is rounded to the nearest integer. This prevents that the description will be an exhaustive list of attributes, which does not sound natural.