Ranking CGANs: Subjective Control over Semantic Image Attributes

Yassir Saquil
Y.Saquil@bath.ac.uk
Kwang In Kim
K.Kim@bath.ac.uk
Peter Hall
P.M.Hall@bath.ac.uk

Visual Computing Group
University of Bath
Bath, UK

Abstract

In this paper, we investigate the use of generative adversarial networks in the task of image generation according to subjective measures of semantic attributes. Unlike the standard (CGAN) that generates images from discrete categorical labels, our architecture handles both continuous and discrete scales. Given pairwise comparisons of images, our model, called RankCGAN, performs two tasks: it learns to rank images using a subjective measure; and it learns a generative model that can be controlled by that measure. RankCGAN associates each subjective measure of interest to a distinct dimension of some latent space. We perform experiments on UT-Zap50K, PubFig and OSR datasets and demonstrate that the model is expressive and diverse enough to conduct two-attribute exploration and image editing.

1 Introduction

Humans routinely order objects based on semantic attributes. For example, people agree on what is "stylish", at least to a sufficient extent for a meaningful communication. However, such subjective concepts are ill defined mathematically, making the building of a computational model an open challenge.

In this paper, we propose an extension to CGAN [33] to address the problem of synthesizing images while controlling subjective measures of semantic attributes. For instance, we want a system that produces shoes images ranked according to the degree they exhibit a semantic attribute *e.g.* "sporty".

In recent works, semantic attributes were defined as categorical labels [6, 7] indicating the presence or the absence of some attributes. Training a conditional generative model, where the labels are assigned to a latent variable, results in a control limited to switching on or off the attribute in the generated images. In contrast, we provide a mapping of semantic attributes onto a continuous subjective scale.

Although humans can order elements in an object class, there is typically no scale associated with the ordering: people can say one pair of shoes is more "sporty" than another pair, but not by how much. To address this problem, a ranking function is learned per attribute which can predict the rank of a novel image using an annotated set of image pairs where each pair is ordered under human supervision.

The key characteristic of our approach is the addition of a *ranking* unit that operates alongside the usual discriminator unit. Both the ranker and the discriminator receive inputs from a generator. As is usual, the role of the discriminator is to distinguish between real and fake images; the role of the ranker is to infer subjective ranking according to the semantic attributes. We call our architecture RankCGAN.

We evaluated RankCGAN on three datasets, shoes (UT-Zap50K) [51], faces (PubFig) [21] and scenes (OSR) [35] datasets. Results in Section 4 show that our model can disentangle multiple

attributes and can keep a correct continuous variation of the attribute's strength with respect to a ranking score, which is not guaranteed with a standard CGAN.

Contributions: In sum, our contributions are: (1) A solution to the problem of rank ordering semantic attributes; (2) A novel conditional generative model that can generate images under semantic attributes that are subject to a global subjective ranking; (3) A training scheme that requires only subjective ranked pairs of images. There is no need for global ranking or to annotate the whole dataset.

2 Related work

Deep Generative Model: There is substantial literature on learning with deep generative models. Early studies were based on RBMs and denoising auto-encoders [10, 11, 44, 48]. Recently, deterministic networks [8, 39, 50] propose architectures for image synthesis. In comparison, stochastic networks rely on a probabilistic formulation of the problem. VAE [19, 41] maximizes a lower bound on the log-likelihood of training data. PixelRNN [34] represents directly the conditional distribution over the pixel space. GANs [8] have the ability to generate sharp image samples with higher resolution.

Several studies have investigated conditional image generation settings. Most of the methods use a supervised approach such as text, attributes and class label conditioning to learn a desired transformation [30, 33, 40, 47]. Additionally, there are works on image-conditioned models, such as style transfer [15, 54], super-resolution [24, 42] and cross-domain image translation [13, 17, 25, 31].

In contrast, fewer works have focused on disentangling latent space representation. InfoGAN [4] learns unsupervisedly semantic features by maximizing mutual information between the latent code and the generated observation. In the spectrum of supervised approaches, VAE [19] was used in DC-IGNs [20] to learn latent codes representing the rendering process of 3D objects, similarly used in Attribute2Image [49] to separately generate the background and the foreground of an image. Attempts to incorporate the adversarial objectives were conducted in VAE [32] and autoencoders [29] settings.

In fader network [22] approach, the attribute is incorporated in the encoded image and trained using binary labels, which requires a careful parametrization. Independently, CFGAN [16] proposes a filtering architecture that enables more variations of the attribute and hence more control options. Finally, RankGAN [26] proposes a method which substitutes the discriminator of GAN by a ranker for generating high-quality natural language descriptions. If we were to project this method on image generation task, the ranker will be modelled to order real image higher than generated ones. But, in our work, the ranker will play a different role: it orders the images according to their present semantic attributes rather than their quality.

Image Editing: Image editing methods have a long history in the research community. Recently, CNN based approaches have shown promising results in image editing tasks such as image colorization [12, 23, 52], filtering [28] and inpainting [36]. These methods use an unsupervised training protocol during the reconstruction of the image, that may not capture important semantic contents.

Handful works used generative models for image editing. iGAN [53] imposes color and shape constraints on an input image, finds the best latent code of GAN satisfying these constraints, and then transfers the generated image motion and color flow during the interpolation to the real image. Similarly, the Neural photo editor [2] proposes an interface for portrait editing using an hybrid VAE/GAN model.

Aiming for high semantic level image editing, the Invertible Conditional GAN [37] trains separately a CGAN on the attributes of interest and an encoder that maps the input image to the latent space of CGAN. Similarly, CFGAN [16] relies on iGAN's [53] approach to estimate the latent variables of an input image, while the Fader network [22] can directly manipulate the attributes using its built-in encoder.

Relative Attributes: Visual concepts can be represented by semantic attributes. In early studies, binary attributes, describing the presence of an attribute, showed excellent performance in object recognition [45] and action recognition [27]. However, relative attributes [35] proposes a better representation that quantifies the strength of an attribute by learning a global ranking function on images

using constraints describing the relative emphasis of attributes. This approach is regarded as solving a learning-to-rank problem where a linear function is learned based on RankSVM [14]. Similarly, RankNet [3] trains a neural network to model the ranking function using gradient descent methods. Also, Criteria Sliders [26] applies semi-supervised learning to learn user-defined ranking functions.

3 Approach

In this section, we outline the generative CGAN [33]; then the discriminative RankNet [3]; and last we describe our contribution by showing how to combine these distinct architectures to build RankCGAN.

3.1 Generation by CGAN

The CGAN [33] is an extension of GAN [8] in a conditional setting. The generative adversarial network (GAN) is a generative model trained using a two-player minmax game. It consists of two networks: a generator G which outputs a generated image $G(z)$ given a latent variable $z \sim p_z(z)$, and a discriminator D which is a binary classifier that outputs a probability $D(x)$ of the input image x being real; that is, sampled from true data distribution $x \sim p_{data}(x)$. The minmax objective is defined as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

In this model, there is no control over the latent space: the prior, $p_z(.)$, on z is often a multivariate normal $\mathcal{N}(0,I)$ or a multivariate uniform $\mathcal{U}(-1,1)$ distribution. Thus sampling generates random images.

CGAN [33] provides some control over the generation process via an additional latent variable, r, that is passed to both the generator and discriminator. The objective of CGAN is expressed as:

$$\min_G \max_D \mathbb{E}_{r,\, x \sim p_{data}(r,\, x)}[\log(D(r, x))] + \mathbb{E}_{z \sim p_z(z),\, r \sim p_r(r)}[\log(1 - D(r, G(r, z)))]. \tag{2}$$

We can now consider the latent space to be partitioned into two parts: a random subspace for z that operates exactly as in standard GAN, and an attribute subspace containing r over which the user has control.

3.2 Discrimination by RankNet

RankNet [3] is a discriminative architecture in the sense that it classifies a pair of inputs, x_i and x_j according to their rank order: $x_i \triangleright x_j$ or $x_i \triangleleft x_j$. As proposed in [23], the architecture learns a ranking function R that maps an input image x to a real ranking score value $R(x)$, such that $x_i \triangleright x_j$ or $R(x_i) > R(x_j)$ means that input x_i is ranked higher than input x_j. The training is supervised by a set of triplets $\{(x_i^{(1)}, x_i^{(2)}, y_i)\}_{i=1}^{P}$ where P is the dataset size, $(x_i^{(1)}, x_i^{(2)})$ is a pair of images and $y_i \in \{0,1\}$ is a binary label indicating whether the image $x_i^{(1)}$ exhibits more of some attribute than $x_i^{(2)}$ or not. The loss function for $(x_i^{(1)}, x_i^{(2)})$ along with the target label y_i is defined as a cross binary entropy:

$$\mathcal{L}_Q(x_i^{(1)}, x_i^{(2)}, y_i) = -y_i \log(p_i) - (1 - y_i)\log(1 - p_i), \tag{3}$$

where the posterior probabilities $p_i = P(x_i^{(1)} \triangleright x_i^{(2)})$ make use of the estimated ranking scores

$$p_i = \mathrm{sig}(R(x_i^{(1)}) - R(x_i^{(2)})) := \frac{1}{1 + e^{-(R(x_i^{(1)}) - R(x_i^{(2)}))}}. \tag{4}$$

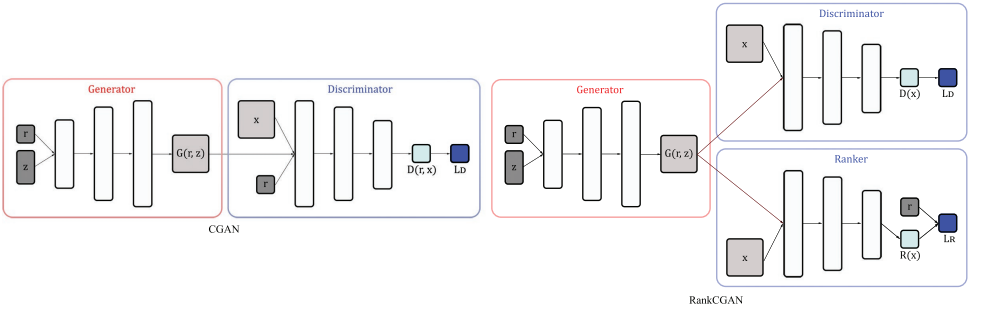

Figure 1: The difference between CGAN and RankCGAN architectures. The dark grey represents the latent variable, light grey represents the observable images, the light blue indicates the output of the discriminator and the ranker, and the dark blue indicates their loss functions.

3.3 Ranking Conditional Generative Adversarial Networks

Recall our aim: to generate images of a particular object class, controlled by one or more subjective attributes. With CGAN and RankNet available, we construct our architecture called RankCGAN. The key novelty of RankCGAN is the addition of a ranking unit where the training scheme puts semantic ordering constraints in the generation process with respect to input latent variables. We note that RankCGAN does not use the matching-aware discriminator [40] as in CGAN because it is trained on continuous controlling variables rather than binary labels.

The Figure 1 illustrates the RankCGAN architecture that is versatile enough to specify points over a line or a plane using one or two semantic attributes respectively, and in principle could operate in three or more dimensions. For simplicity's sake, we open our description with the one-dimensional case, followed by a generalization to n dimensions. Finally, we augment RankCGAN with an encoder that allows users to specify the subjective degree of semantic attributes, and in that way control image editing.

3.3.1 The one-attribute case

The generator G takes two inputs, $z \sim \mathcal{N}(0, I)$ the unconditional latent vector, and $r \sim \mathcal{U}(-1, 1)$ the latent variable controlling the attribute. The generator outputs an image $x = G(r, z)$. This image is input not just to a discriminating unit, $D(x)$, as in CGAN, but also to a ranking unit $R(x)$; as seen in Figure 1. Consequently, the architecture has three sets of parameters Θ_G, Θ_D, and Θ_R for the generator, discriminator, and ranker, respectively. As for RankNet, let $\{(x_i^{(1)}, x_i^{(2)}, y_i)\}_{i=1}^P$ be pairwise comparisons and $\{x_i\}_{i=1}^N$ an image dataset of size N. These datasets are used in a supervised mini-batch training scheme of size B by defining three loss functions, one for each unit: generation ($\mathcal{L}_G$), discrimination ($\mathcal{L}_D$), and ranking ($\mathcal{L}_R$).

$$\mathcal{L}_G(I) = -\frac{1}{B}\sum_{i=1}^{B} \log[D(I_i)], \tag{5}$$

such that $\{I_i\}_{i=1}^B = \{G(r_i, z_i)\}_{i=1}^B$.

$$\mathcal{L}_D(I) = -\frac{1}{B}\sum_{i=1}^{B} (t\log[D(I_i)] + (1-t)\log[1-D(I_i)]), \tag{6}$$

where $t = \begin{cases} 1 & \text{if } \{I_i\}_{i=1}^{B} = \{x_i\}_{i=1}^{B}, \\ 0 & \text{if } \{I_i\}_{i=1}^{B} = \{G(r_i, z_i)\}_{i=1}^{B}. \end{cases}$

$$\mathcal{L}_R(I^{(1)}, I^{(2)}, l) = \frac{2}{B}\sum_{i=1}^{B/2}\mathcal{L}_Q(I_i^{(1)}, I_i^{(2)}, l_i) = -\frac{2}{B}\sum_{i=1}^{B/2}(l_i \log[\text{sig}(R(I_i^{(1)}) - R(I_i^{(2)}))] + (1-l_i)\log[1-(\text{sig}(R(I_i^{(1)}) - R(I_i^{(2)})))]), \tag{7}$$

with

$$\left\{(I_i^{(1)}, I_i^{(2)}, l_i)\right\}_{i=1}^{B/2} = \begin{cases} \left\{(x_i^{(1)}, x_i^{(2)}, y_i)\right\}_{i=1}^{B/2}, \text{ 'for real images';} \\ \left\{(G(r_i^{(1)}, z_i^{(1)}), G(r_i^{(2)}, z_i^{(2)}), f(r_i^{(1)}, r_i^{(2)}))\right\}_{i=1}^{B/2}, \text{ 'for synthesised images';} \end{cases}$$

$$f(r_i^{(1)}, r_i^{(2)}) = \begin{cases} 1 & \text{if } r_i^{(1)} > r_i^{(2)}, \\ 0 & \text{else.} \end{cases} \tag{8}$$

The Algorithm 1 defines the adversarial training of RankCGAN using the loss functions. The hyper-parameter λ controls the contribution of the ranker-discriminator during the updates of the generator.

3.3.2 Multiple semantic attributes

An interesting feature of RankCGAN is the model extension to multiple attributes. The architecture design and the training procedure remain intact, the only differences are the incorporation of new latent variables that control the additional attributes and the structure of the ranker which outputs a vector of ranking score with respect to each attribute. There are two ways of designing a multi-attribute ranker, either using a separate ranking layer for each attribute, or a single ranking layer shared between all the attributes. Let $\{(x_i^{(1)}, x_i^{(2)}, \mathbf{y}_i)\}_{i=1}^{P}$ be pairwise comparisons where $\mathbf{y}_i$ is a vector of binary labels indicating whether $x_i^{(1)} \triangleright x_i^{(2)}$ or not with respect to all S attributes. We define the loss function of ranking:

$$\mathcal{L}_R(I^{(1)}, I^{(2)}, l) = \sum_{j=1}^{S}\mathcal{L}_{R_j}(I^{(1)}, I^{(2)}, l_j), \tag{9}$$

$\mathcal{L}_{R_j}(I^{(1)}, I^{(2)}, l_j)$ is the ranking loss with respect to the attribute j, defined similarly to Equation (7):

$$\mathcal{L}_{R_j}(I^{(1)}, I^{(2)}, l_j) = -\frac{2}{B}\sum_{i=1}^{B/2}(l_{ij} \log[\text{sig}(R(I_i^{(1)}) - R(I_i^{(2)}))] + (1-l_{ij})\log[1-(\text{sig}(R(I_i^{(1)}) - R(I_i^{(2)})))]), \tag{10}$$

such that

$$\left\{(I_i^{(1)}, I_i^{(2)}, l_{ij})\right\}_{i=1}^{B/2} = \begin{cases} \left\{(x_i^{(1)}, x_i^{(2)}, y_{ij})\right\}_{i=1}^{B/2}, \text{ 'for real images';} \\ \left\{(G(r_{ij}^{(1)}, z_i^{(1)}), G(r_{ij}^{(2)}, z_i^{(2)}), f(r_{ij}^{(1)}, r_{ij}^{(2)}))\right\}_{i=1}^{B/2}, \text{ 'for synthesised images';} \end{cases}$$

$$f(r_{ij}^{(1)}, r_{ij}^{(2)}) = \begin{cases} 1 & \text{if } r_{ij}^{(1)} > r_{ij}^{(2)}, \\ 0 & \text{else,} \end{cases} \tag{11}$$

with y_{ij} j-th element in $\mathbf{y}_i$ and r_{ij} i-th input in the mini-batch, associated to the j-th attribute latent variable.

Algorithm 1 RankCGAN

Set the learning rate η, the batch size B, and the training iterations S
Initialize each network parameters Θ_D,Θ_R,Θ_G
Data Images Set $\{x_i\}_{i=1}^{N}$, Pairs Set $\left\{(x_i^{(1)}, x_i^{(2)}, y_i)\right\}_{i=1}^{P}$
for n=1 **to** S **do**
Get real images mini-batches:
$x_{real} = \{x_i\}_{i=1}^{B}, x_{real}^{(pair)} = \left\{(x_i^{(1)}, x_i^{(2)}, y_i)\right\}_{i=1}^{B/2}$
Get fake images mini-batches:
$x_{fake} = \{G(r_i, z_i)\}_{i=1}^{B}, x_{fake}^{(pair)} = \left\{\left(G(r_i^{(1)}, z_i^{(1)}), G(r_i^{(2)}, z_i^{(2)}), f(r_i^{(1)}, r_i^{(2)})\right)\right\}_{i=1}^{B/2}$
Update the discriminator D:
$\Theta_D \leftarrow \Theta_D - \eta \cdot \left(\frac{\partial L_D(x_{real})}{\partial \Theta_D} + \frac{\partial L_D(x_{fake})}{\partial \Theta_D}\right)$
Update the ranker R:
$\Theta_R \leftarrow \Theta_R - \eta \cdot \frac{\partial L_R(x_{real}^{(pair)})}{\partial \Theta_R}$
Update the generator G:
$\Theta_G \leftarrow \Theta_G - \eta \cdot \left(\frac{\partial L_G(x_{fake})}{\partial \Theta_G} + \lambda \cdot \frac{\partial L_R(x_{fake}^{(pair)})}{\partial \Theta_G}\right)$
end for

3.4 An Encoder for Image Editing

The image editing task consists of inferring the latent variables (r, z) of an image x and generating the desired image by manipulating r. Since GAN lacks an inference mechanism, we use latent variables estimation for such task. The approach [37] consists of creating a dataset of size M from the generated images and their latent variables $\{r_i, z_i, G(r_i, z_i)\}_{i=1}^{M}$ and training the encoders E_r, E_z which encode to r and z on this dataset. Their loss functions are defined in the mini-batch setting as follows:

$$\mathcal{L}_{Ez} = \frac{1}{B}\sum_{i=1}^{B} \| z_i - E_z(G(r_i, z_i)) \|_2^2, \qquad \mathcal{L}_{Er} = \frac{1}{B}\sum_{i=1}^{B} \| r_i - E_r(G(r_i, z_i)) \|_2^2. \tag{12}$$

To reach a better estimation of z and r we can use the manifold projection method proposed in [53] which consists of solving the following problem:

$$r^*, z^* = \underset{r,\, z}{\operatorname{argmin}} \| x - G(r, z) \|_2^2. \tag{13}$$

Unfortunately, this problem is non-convex, so that obtained estimates for r^* and z^* are strongly contingent upon the initial values of r, z; good initial values are provided by the encoders E_r, E_z.

4 Empirical Results

4.1 Datasets

We used three datasets that provide relative attributes: UT-Zap50K [51], PubFig [21], and OSR [35].

UT-Zap50K [51] consists of 50,025 shoes images with white backgrounds and same orientation rescaled to 64×64 for GAN training purpose. The annotations for pairwise comparisons comprise 4 attributes (sporty, pointy, open, comfortable). We defined the "black" attribute by comparing the

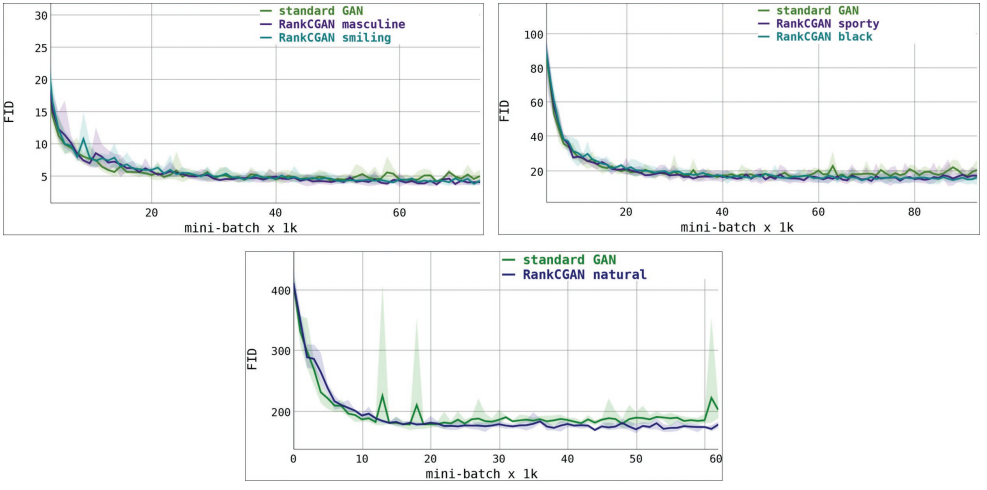

Figure 2: Mean FID (solid line) surrounded by a shaded area bounded by the maximum and the minimum over 5 runs for GAN/RankCGAN on PubFig, OSR and UT-Zap50K datasets. **Left:** PubFig with two RankCGANs trained on "masculine" and "smiling" attributes, starting at 3k mini-batch for better visualisation. **Middle:** UT-Zap50K with two RankCGANs trained on "sporty" and "black" attributes, also starting at 3k mini-batch. **Right:** OSR with one RankCGAN trained on "natural" attribute.

color histogram of images. We relied on coarse pairs collection in our experiments because they are easy to visually discern. It contains 1,500 to 1,800 ordered pairs per each attribute.

PubFig dataset [21] consists of 15,738 downloaded face images rescaled to 64×64. A subset of PubFig dataset is used to build a relative attributes dataset [1] containing 900 facial images of 60 categories and 29 attributes. The ordering of samples is annotated in a category level with respect to an attribute. We used 50,000 pairwise image comparisons from the ordered categories.

Finally, Outdoor Scene Recognition dataset (OSR) [35] consists of 2,688 images rescaled to 64×64 from 8 scene categories and 6 attributes. Similar to PubFig, the attributes are defined in a category level, which enable to create 50,000 pairwise image comparisons to train the ranker on the desired attribute.

4.2 Implementation

Our RankCGAN is built on top of DCGAN [38] implementation with the same hyperparameters setting and structure of the discriminator D and the generator G. Besides, we modelled the encoders E_r, E_z, and the ranker R with the same architecture of the discriminator D, except for the last sigmoid layer. We set the hyperparameter $\lambda = 1$, learning rate to 0.0002, mini-batch of size 64, and trained the networks using mini-batch stochastic gradient descent (SGD) with Adam optimizer [18]. We trained the networks on UT-Zap50K and PubFig datasets for 300 epochs and on OSR dataset for 1,500 epochs.

4.3 Quantitative Results

The Fréchet Inception Distance (FID) [9] is a recent evaluation method for GAN models. It consists of estimating the mean and covariance of a multivariate Gaussian distribution for both real and generated images using Inception Net pool 3 layer for feature extraction. The Fréchet Distance between these two Gaussians is used to quantify the quality of the generated images. In figure 2, we compare RankCGAN with the standard GAN on all datasets by measuring the quality of generated images. We calculate FID on every 1,000 mini-batch iterations such that the size of generated images is equal to the size of the

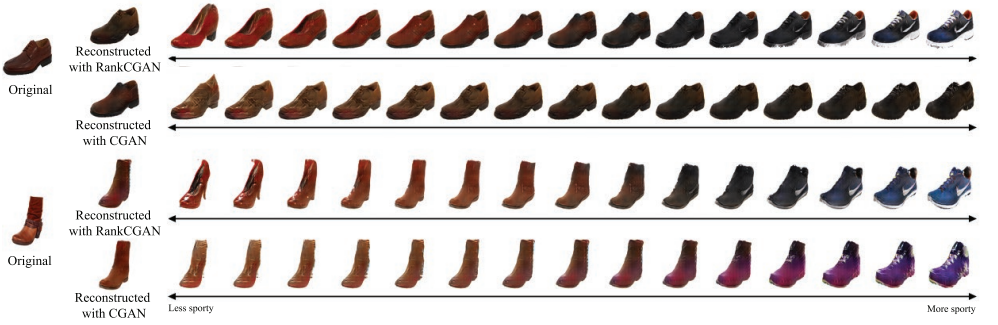

Figure 3: Examples of the interpolation of similar images with respect to "sporty" attribute using RankCGAN (rows 1, 3) and CGAN (rows 2, 4).

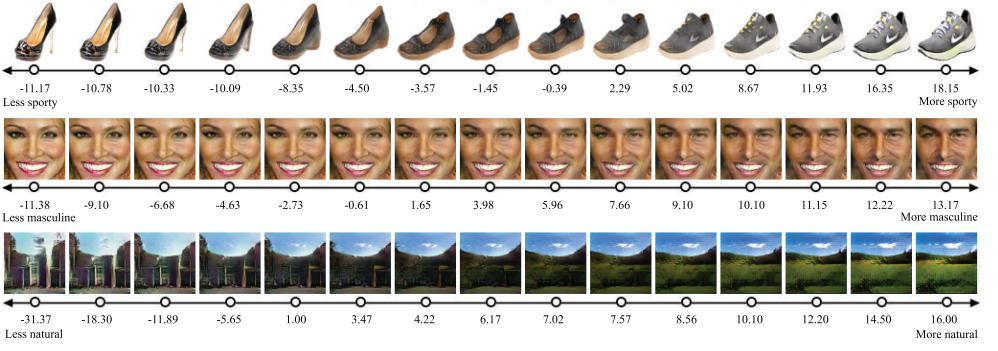

Figure 4: Generated shoe (top), face (middle), and scene (bottom) images associated with their ranking scores using "sporty", "masculine" and "natural" attributes respectively.

used dataset. We notice that the standard GAN is a little faster in the beginning but eventually RankCGAN achieves slightly better or equal performance to GAN. The FID scores are higher in OSR dataset due to the difficulty to generate its images, while the FID scores are lower on UT-Zap50K and PubFig datasets which show that our model generative capability and quality are tied to the standard GAN.

4.4 Qualitative Results

We would like an experiment that tests the added value of RankCGAN in comparison with CGAN. Unfortunately, conducting such experiment is not straightforward due to the fact that RankCGAN requires ordering pairs for the training, whereas CGAN requires binary labels.

To enable a comparison, we follow [46] method in mapping pairwise ordering to binary labels in order to train CGAN on all datasets. It consists of training a ranker identical to RankNet to map each real image in the dataset to a real score. Then, all images with a negative score are said to not have the semantic property, while those with a positive score are said to possess it. The proportion of images with positive and negative scores in all the datasets is balanced around the threshold zero.

Figure 3 shows the qualitative results. We used the encoders E_r, E_z to estimate the latent variables r and z of a given real image. The results suggest that RankCGAN is capable of spanning a wider subjective interval than CGAN. Evidence for this is seen in the extreme ends of each interval. The top line (second row) of CGAN fails to reach shoes that can reasonably be called "sporty", while the bottom line (fourth row) fails to include a high-heel at the "not sporty" extremity. In contrast,

Figure 5: **Left:** Example of two-attributes interpolation on shoe and face images using ("sporty","black") and ("masculine","smiling") attributes. **Right:** Examples of "smiling" attribute transfer task. The latent variable r of reference images is estimated and then transferred to the target image.

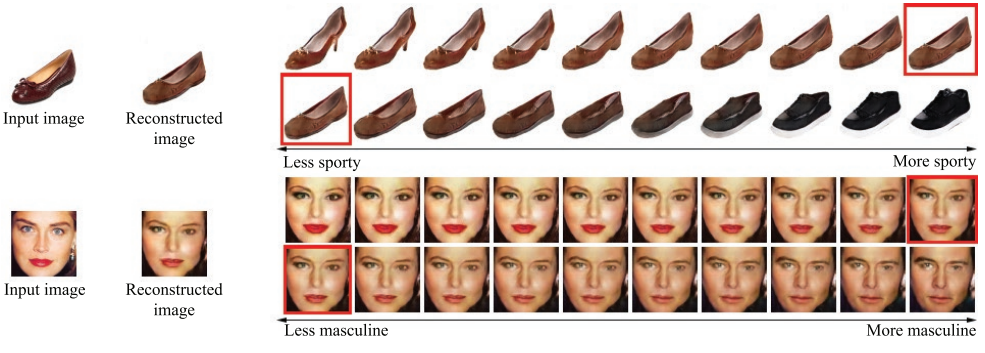

Figure 6: Examples of image editing task, where the latent variables are estimated for an input image to perform the editing with respect to the "sporty" attribute (top) and "masculine" attribute (bottom).

RankCGAN reaches a desirable shoe across the scale in both cases (first and third rows), we see a dress shoe at one end and a sporty shoe at the other.

4.5 Application: Image Generation

In order to demonstrate image generation for RankCGAN, we choose a single and pair semantic attributes that span a line and plane respectively and we held the noise vector z constant.

Figure 4 shows how the generated images vary with respect to the value of some subjective variable. We note that plausible shoe and face images are generated at every point, scene images are more difficult to generate, but our images are reasonable and progress from "urban" to "natural" in a pleasing manner. In all cases, the semantic attribute changes smoothly over subjective scale.

Figure 5 (left) illustrates the image generation using two-attributes. We trained RankCGAN on multiple attributes using the separate ranking layers strategy because the shoes dataset provides different pairs of images with respect to a specific semantic attribute. We used ("sporty", "black") and ("masculine", "smiling") attributes. In both cases plausible images are generated, and the progression in both directions smoothly adheres to the subjective attributes in question. We invite the reader to the supplementary material for additional application results on more attributes.

4.6 Application: Image Editing

The core of image editing is to map a given image onto the subjective scale by estimating its latent variables r, z that produce a reconstructed image, and then move along the scale, one way or the other.

In figure 6, we show image editing on shoes and faces, using the "sporty" and "masculine" attributes respectively. The reconstructed images are framed in red, and the images generated with subjectively less of the chosen attribute are shown on the top line, while the images generated to have subjectively more are on the bottom line. In both cases the noise vector z was held constant.

4.7 Application: Semantic Attribute Transfer

Our final application is semantic attribute transfer. The idea is to extract the subjective measure of a semantic attribute from one image, and apply that measure to another. Formally, we quantify the conditional variable r of the source image using the encoder E_r, then edit the target image with the new estimated semantic value. Figure 5 (right) shows some examples. The reference images are ordered, left to right, by increasing the subjective level of "smiling". The corresponding semantic value is then used in conjunction with each of the target images to generate a new expression for the person in the picture.

5 Discussion and Conclusion

We introduced RankCGAN, a novel GAN architecture that synthesises images with respect to semantic attributes defined relatively using a pairwise comparisons annotation. We showed through experiments that the design and training scheme of RankCGAN enable latent semantic variables to control the attribute strength in the generated images using a subjective scale.

Our proposed model is generic in the sense that it can be integrated into any extended CGAN model. It follows that, our model's generative power is tied to that of the GAN in terms of the quality of generated images, the diversity of the model and the evaluation methods.

Possible extensions to this study consist of incorporating a filtering architecture, CFGAN [16], to enhance the RankCGAN controllability and also incorporating an encoder in the RankCGAN to perform an end-to-end training in order to improve image editing and attribute transfer tasks.

6 Acknowledgements

We would like to thank James Tompkin for initial discussions about the main ideas of the paper. Yassir Saquil thanks the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 665992 and the UK's EPSRC Centre for Doctoral Training in Digital Entertainment (CDE), EP/L016540/1. Kwang In Kim thanks EPSRC EP/M00533X/2 and RCUK EP/M023281/1.

References

[1] Arijit Biswas and Devi Parikh. Simultaneous active learning of classifiers & attributes via relative feedback. In *CVPR*, 2013.

[2] Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. In *ICLR*, 2017.

[3] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[4] Xi Chen, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*. 2016.

[5] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015.

[6] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[7] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. In *NIPS*, 2007.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*. 2014.

[9] 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 *NIPS*. 2017.

[10] Geoffrey E. Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[11] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[12] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. In *SIGGRAPH*, 2016.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[14] Thorsten Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[16] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. Generative attribute controller with conditional filtered generative adversarial networks. In *CVPR*, 2017.

[17] Taeksoo Kim, Byoungjip Kim, Moonsu Cha, and Jiwon Kim. Unsupervised visual attribute transfer with reconfigurable generative adversarial networks. *CoRR*, abs/1707.09798, 2017. URL http://arxiv.org/abs/1707.09798.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[20] Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015.

[21] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.

[22] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *CoRR*, abs/1706.00409, 2017. URL http://arxiv.org/abs/1706.00409.

[23] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016.

[24] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[25] Xiaodan Liang, Hao Zhang, and Eric P. Xing. Generative semantic manipulation with contrasting GAN. *CoRR*, abs/1708.00315, 2017. URL http://arxiv.org/abs/1708.00315.

[26] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming ting Sun. Adversarial ranking for language generation. In *NIPS*. 2017.

[27] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR*, 2011.

[28] Sifei Liu, Jin-shan Pan, and Ming-Hsuan Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *ECCV*, 2016.

[29] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. In *ICLR*, 2016.

[30] Elman Mansimov, Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *ICLR*, 2016.

[31] Xudong Mao, Qing Li, and Haoran Xie. AlignGAN: Learning to align cross-domain images with conditional generative adversarial networks. *CoRR*, abs/1707.01400, 2017. URL http://arxiv.org/abs/1707.01400.

[32] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *NIPS*. 2016.

[33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

[34] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

[35] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, 2011.

[36] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[37] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. In *NIPS Workshop on Adversarial Training*, 2016.

[38] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[39] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *NIPS*. 2015.

[40] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[42] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised MAP inference for image super-resolution. In *ICLR*, 2017.

[43] Yaser Souri, Erfan Noury, and Ehsan Adeli. Deep relative attributes. In *ACCV*, 2016.

[44] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.

[45] Ran Tao, Arnold W. M. Smeulders, and Shih-Fu Chang. Attributes and categories for generic instance search from one example. In *CVPR*, 2015.

[46] James Tompkin, Kwang In Kim, Hanspeter Pfister, and Christian Theobalt. Criteria sliders: Learning continuous database criteria via interactive ranking. In *BMVC*, 2017.

[47] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

[48] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[49] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016.

[50] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*. 2015.

[51] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *CVPR*, 2014.

[52] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[53] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.