DesIGN: Design Inspiration from Generative Networks

Othman Sbai[1,2] Mohamed Elhoseiny[1] Antoine Bordes[1] Yann LeCun[1,3] Camille Couprie[1]

[1] Facebook AI Research
[2] École des Ponts, UPE
[3] New York University

Abstract

Can an algorithm create original and compelling fashion designs to serve as an inspirational assistant? To help answer this question, we design and investigate different image generation models associated with different loss functions to boost creativity in fashion generation. The dimensions of our explorations include: (i) different Generative Adversarial Networks architectures that start from noise vectors to generate fashion items, (ii) a new loss function that encourages creativity, and (iii) a generation process following the key elements of fashion design (disentangling shape and texture makers). A key challenge of this study is the evaluation of generated designs and the retrieval of best ones, hence we put together an evaluation protocol associating automatic metrics and human experiment studies that we hope will help ease future research. We show that our proposed creativity loss yields better overall appreciation than the one employed in Creative Adversarial Networks. In the end, about 61% of our images are thought to be created by human designers rather than by a computer while also being considered original.

1. Introduction

Artificial Intelligence (AI) research has been making huge progress in the machine's capability of human level understanding across the spectrum of perception, reasoning and planning [14, 1, 28].

Another key yet still relatively understudied direction is creativity where the goal is for machines to generate original items with realistic, aesthetic and/or thoughtful attributes, usually in artistic contexts. We can indeed imagine AI to serve as inspiration for humans in the creative process and also to act as a sort of creative assistant able to help with more mundane tasks, especially in the digital domain. Previous work has explored writing pop songs [3], imitating the styles of great painters [9, 7] or doodling sketches [12] for instance. However, it is not clear how *creative* such attempts can be considered since most of them mainly tend to mimic training samples without expressing much originality.

Figure 1: Training generative adversarial models with appropriate losses leads to realistic and creative 512×512 fashion images.

Creativity is a subjective notion that is hard to define and evaluate, and even harder for an artificial system to optimize for. Colin Martindale put down a psychology based theory that explains human creativity in art [22] by connecting creativity or acceptability of an art piece to novelty with *"the principle of least effort"*. As originality increases, people like the work more and more until it becomes too novel and too far from standards to be understood. When this happens, people do not find the work appealing anymore because a lack of understanding and of realism leads to a lack of appreciation. This behavior can be illustrated by the Wundt curve that correlates the arousal potential (i.e. novelty) to hedonic responses (e.g. likability of the work) with an inverted U-shape curve.

Earlier works on computational creativity for generating paintings have been using genetic algorithms [21, 20, 23] to create new artworks by starting from existing human-generated ones and gradually altering them using pixel transformation functions. There, the creativity is guided by pre-defined fitness functions that can be tuned to trade-off novelty and realism. For instance, for generating portraits, DiPaola and Gabora [6] define a family of fitness functions based on specific painterly rules for portraits that can guide creation from emphasizing resemblance to existing paintings to promoting abstract forms.

Generative Adversarial Networks (GANs) [10, 25] show a great capability to generate realistic images from scratch without requiring any existing sample to start the generation from. They can be applied to generate artistic content, but their intrinsic creativity is limited because of their training process that encourages the generation of items close to the training data distribution; hence they show limited originality and overall creativity. Similar conservative behavior can be seen in recent deep learning models for music generation where the systems are also mostly trained to reproduce pattern from training samples, like Bach chorales [13]. Creative Adversarial Networks (CANs) [8] have then been proposed to adapt GANs to generate creative content (paintings) by encouraging the model to deviate from existing painting styles. Technically, CAN is a Deep Convolutional GAN (DCGAN) model [25] associated with an entropy loss that encourages novelty against known art styles. The specific application domain of CANs (art paintings) allows for very abstract generations to be acceptable but, as a result, does reward originality a lot without judging much how such enhanced creativity can be mixed with realism and standards.

In this paper we study how AI can generate creative samples for fashion. Fashion is an interesting domain because designing original garments requires a lot of creativity but with the constraints that items must be wearable. For decades, fashion designers have been inventing styles and innovating on shapes and textures while respecting clothing standards (dimensions, etc.) making fashion a great playground for testing AI creative agents while also having the potential to impact everyday life.

In contrast to most generative models works, the creativity angle we introduce makes us go beyond replicating images seen during training. We choose fashion design generation due to its impact in our life and that it opens the door for breaking creativity into design elements (shape and texture in our case), which is a novel aspect of our work in contrast to CANs.

More specifically, this work explores various architectures and losses that encourage GANs to deviate from existing fashion styles covered in the training dataset, while still generating realistic pieces of clothing without needing any image as input. In the fashion industry, the design process is traditionally organized around the collaboration between pattern makers, responsible for the material, the fabric and the texture, and sample makers, working on the shape models. As detailed in Table 1, we follow a similar process in our exploration of losses and architectures. We compare the relative impact of both shape and texture dimensions on final designs using a comprehensive evaluation that jointly assesses novelty, likeness and realism using both adapted automatic metrics, and humans subject experiments performed on Mechanical Turk. To the best of our knowledge, this work is the first attempt at incorporating creative fashion generation by explicitly relating it to its design elements.

Our contributions are the following:

- We are the first to propose a creativity loss on image generation of fashion items with a specific conditioning of texture and shape, learning a deviation from existing ones, constituting *a novel application*.
- In addition, we are the first to propose the *better leading results, more general*, MCE criterion for learning to deviate from existing shapes and textures.
- We re-purposed automatic entropy based evaluation criteria for assessment of fashion items in terms of texture and shape; The correlations between the automatic metrics that we proposed and our human study allowed us to *draw some conclusions with useful metrics revealing human judgment*.
- We proposed a concrete solution to make our Style GAN model work in a non-deterministic way, and trained it with creative losses, resulting in a *novel and powerful model*.

As illustrated in Fig. 1, our best models manage to generate realistic images with high resolution 512×512 using a relatively small dataset (about 5000 images). More than 60% of our generated designs are judged as being created by a human designer while also being considered original, showing that an AI could offer benefits serving as an efficient and inspirational assistant.

2. Related work

There has been a growing interest in generating images using convolutional neural networks and adversarial training, given their ability to generate appealing images unconditionally, or conditionally like from text, class labels, and for paired and unpaired image translations [35]. Introduced by Goodfellow et al. [10], GANs allow image generation

Architecture	Creativity Loss	Design Elements
DCGAN [5]	CAN [8]	Texture
StackGAN (ours)	MCE (ours)	Shape
StyleGAN (ours)		Shape and Texture
		None

Table 1: Dimensions of our study (19 combinations)

from random numbers using two networks trained simultaneously: a generator is trained to fool an adversarial network by generating images of increasing realism. The initial resolution of generated images was 32×32.

From this seminal work, some progresses were achieved in generating higher resolution images. Using a cascade of convolutional networks Denton et al. [5] increased it up to 96×96. Going deeper in their network architectures, Radford et al. [25] demonstrated the usefulness of the features learned by the adversarial network, also called the discriminator, in unsupervised image classification tasks. Odena et al. introduced Auxiliary Classifier GANs [24] by adding a label input in addition to the noise and training the discriminator to classify the synthesized 128×128 images. Using text inputs, Reeds et al. [26], and then Zhang et al. [33] allowed the generative network to focus on the area of semantic interest and generate photo-realistic 256×256 images. Recently, impressive 1024×1024 results were obtained using a progressive growing of the generator and discriminator networks [18], by training models during several tens of days.

Neural style transfer methods [9, 17] opened the door to the application of existing styles on clothes [4], the difference with generative models being the constraint to start from an existing image in input. Isola et al. [16] relax this constraint partly by starting from a binary image of edges, and present some generation of handbags images. Another way to control the appearance of the result is to enforce some similarity between input texture patch and the resulting image [32]. Using semantic segmentation and large datasets of people wearing clothes, the works of Zhu et al. [36] and Lassner et al. [19] are built to generate full bodies images and are conditioning their outputs either on text descriptions, color or pose information.

In this work, we are interested in exploring the creativity of generative models and focus on presenting results using only random or shape masks as inputs to leave freedom for a full exploration of GANs creative power.

3. Models: architectures and losses

Table 1 summarizes our models and losses exploration. Let us consider a dataset $\mathcal{D}$ of N images. Let x_i be a real image sample and z_i a vector of n of real numbers sampled from a normal distribution. In practice $n = 100$.

3.1. GANs

As in [10, 25], the generator parameters θ_G are learned to compute examples classified as real by D:

$$\min_{\theta_G} \mathcal{L}_{G \text{ real/fake}} = \min_{\theta_G} \sum_{z_i \in \mathbb{R}^n} \log(1 - D(G(z_i))).$$

The discriminator D is trained to classify the true samples as 1 and the generated ones as 0:

$$\min_{\theta_D} \mathcal{L}_{D \text{ real/fake}} = \min_{\theta_D} \sum_{\substack{x_i \in \mathcal{D} \\ z_i \in \mathbb{R}^n}} -\log D(x_i) - \log(1 - D(G(z_i))). \quad (1)$$

3.2. GANs with classification loss

Following [24], we use shape and texture labels to learn a shape classifier and a texture classifier in the discriminator. Adding those labels improves over the plain model and stabilizes the training for larger resolution. Let us define the texture and shape integer labels of an image sample x by $\hat{t}$ and $\hat{s}$ respectively. We are adding to the discriminator network either one branch for texture D_t or shape D_s classification or two branches for both shape and texture classification $D_{\{t,s\}}$. In the following section, for genericity, we employ the notation $D_{b,k}$, designating the output of the classification branch b for class $k \in \{1, ..., K\}$, where K is the number of different possible classes of the considered branch (shape or texture). We add to the discriminator loss the following classification loss:

$$\mathcal{L}_D = \lambda_{D_r} \mathcal{L}_{D \text{ real/fake}} + \lambda_{D_b} \mathcal{L}_{D\text{classif}} \quad \text{with}$$
$$\mathcal{L}_{D \text{ classif}} = -\sum_{x_i \in \mathcal{D}} \log(\text{softmax}(D_{b,\hat{c}_i}(x_i))),$$

where $\hat{c}_i$ is the label of the image x_i for branch b.

3.3. Creativity losses

Because GANs learn to generate images very similar to the training images, we explore ways to make them deviate from this replication by studying the impact of two additional losses for the generator. The final loss of the generator that are optimizing jointly with $\mathcal{L}_D$ is:

$$\mathcal{L}_G = \lambda_{G_r} \mathcal{L}_{G \text{ real/fake}} + \lambda_{G_e} \mathcal{L}_{G \text{ creativity}}$$

In the following, we denote models employing a creativity loss by Creative Adversarial Networks.

Binary cross entropy loss (CAN [8]) Given the adversarial network's branch D_c trained to classify different textures or shapes, we can use the CAN loss $\mathcal{L}_{\text{CAN}}$ as $\mathcal{L}_{G\text{ creativity}}$ to create a new style that confuses D_b:

$$\mathcal{L}_{\text{CAN}} = -\sum_{x_i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{K} \log(\sigma(D_{b,k}(x_i))) + \frac{K-1}{K} \log(1 - \sigma(D_{b,k}(x_i)))$$

where σ denotes the sigmoid function.

Holistic CAN loss exploiting multi-class cross entropy We propose to use as $\mathcal{L}_{G\text{ creativity}}$ the Multi-class Cross Entropy (MCE) loss between the class prediction of the discriminator and the uniform distribution. The goal is for the generator to make the generations hard to classify by the discriminator.

$$\begin{aligned}\mathcal{L}_{\text{MCE}} &= -\sum_i \frac{1}{K} \log \text{softmax}(D_b(x_i)) \\ &= -\sum_i \frac{1}{K} \log \left(\frac{e^{D_{b,\hat{c}_i}(x_i)}}{\sum_{k=1}^{K} e^{D_{b,k}(x_i)}} \right)\end{aligned}$$

We denote this criterion as CAN(H), for Creative Adversarial Network with Holistic loss. In contrast to the CAN loss that treats every classification independently, the CAN(H) loss should better exploit the class information in a global way. Both MCE and sum of binary cross entropies losses encourage deviation from existing categories (shapes/textures here). Since the generator is trained with generated images that are not associated with any label, the CAN loss can not contrast against negatives and hence merely look at each dimension interdependently encouraging it to get close to $\frac{1}{K}$. Our holistic loss is in fact equivalent to a KL loss between a uniform distribution and the softmax output, since the entropy term of the uniform distribution is constant. It is more meaningful conceptually to define the deviation over the joint distribution over the categories which is what we modeled by the KL divergences between the softmax probabilities and the prior (defined as uniform in our case), as it also opens the door to study other divergence measures. Finally, the CAN(H) loss requires less compute comparing to CAN.

3.4. Network architectures

We experiment using three architectures : modified versions of the DCGAN model[25], StackGANs [33] with no text conditioning, and our proposed styleGAN.

DCGAN The DCGAN generator's architecture was only modified to output 256×256 or 512×512 images. The discriminator architecture also includes these modifications and contains additional classification branches depending on the employed loss function.

Unconditional StackGAN Conditional StackGAN [33] has been recently proposed to generate 256×256 images conditioned on captions for mainly birds and flowers. The method first generates a low resolution 64×64 image conditioned on text. Then, the generated image and the text are tiled on a $16 \times 16 \times 512$ feature map extracted from the 64×64 generated image to compute the final 256×256 image. We adapted the architecture by removing the conditional units (i.e. the text) but realized that it did not perform well for our application. The upsampling in [33] was based on nearest neighbors which we found ineffective in our setting. Instead, we first generate a low resolution image from normal noise using a DCGAN architecture [25], then conditioning on it, we build a higher resolution image of 256×256 with a generator inspired from the pix2pix architecture with 4 residual blocks [16]. The upsampling was performed using transposed convolutions. The details of the architecture are provided in the appendix.

StyleGAN: Conditioning on masks To grant more control on the design of new items and get closer to standard fashion processes where shape and texture are handled by different specialists, we also introduce a model taking binary masks representing a desired shape in input. Since that even for images on white background a simple threshold fails to extract accurate masks, we computed them using a graph based random walk algorithm [11].

In the StyleGAN model, a generator is trained to compute realistic images from a mask input and noise representing style information (see Figure 2), while a discriminator is trained to differentiate real from fake images. We use the same discriminator architecture as in DCGAN with classifier branches that learn shape and texture classification on the real images on top of predicting real/fake discrimination.

Previous approaches of image to image translation such as pix2pix [16] and CycleGAN [35] create a deterministic mapping between an input image to a single corresponding one, i.e. edges to handbags for example or from one domain to another. This is due to the difficulty of training a generator with two different inputs,namely mask and style noise, and making sure that no input is being neglected. In order to allow sampling different textures for the same shape as a design need, we avoid this deterministic mapping by enforcing an additional ℓ_1 loss on the generator:

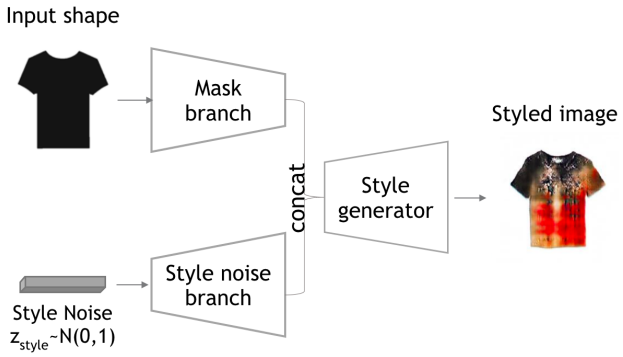

Figure 2: From the segmented mask of a fashion item and different random vector z, our StyleGAN model generates different styled images.

$$\mathcal{L}_{rec} = \sum_{i} \sum_{p \in \mathcal{P}} |G(m_{i,p}, z_i = 0) - m_{i,p}|, \qquad (2)$$

where $m_{i,p}$ denotes the mask of a sample image x_i at pixel p, and $\mathcal{P}$ denotes the set of pixels of m_i.

This loss reconstructs the input mask in case of null input z (i.e. zeros) and hence ensures the impact of the weights in the noise upsampling branch before concatenation as shown in Fig. 2.

4. Evaluation metrics

Training generative models on fashion datasets generates impressive designs mixed with less impressive ones, requiring some effort of visual cherry-picking. We propose automated criteria to evaluate trained models and compare different architectures and loss setups. We study in Section 5 how these automatic metrics correlate with the human evaluation of generated images.

4.1. Automatic evaluation metrics

Evaluating the diversity and quality of a set of images has been tackled by scores such as the inception score and variants like the AM score [34]. We adapt both of them for two labels specific to fashion design (shape and texture) and supplement them by a mean nearest neighbor distance.

Our final set of automatic scores contains 5 metrics :

- shape score and texture score, each based on a Resnet-18 classifier of (shape or texture respectively);
- shape AM score and texture AM score, based on the output of the same classifiers;
- mean distance to 10 nearest neighbors score.

4.1.1 Scores based on classification entropies

Inception-like scores The Inception score [27, 31] was introduced as a metric to evaluate the diversity and quality of generations, with respect to the output of a considered classifier [29]. For evaluating N samples $\{x\}_1^N$, it is computed as

$$I_{score}(\{x\}_1^N) = \exp(\mathbb{E}[KL(C(x)||\mathbb{E}[C(x)])]),$$

where $C(x)$ is the softmax output of the trained classifier C, originally the Inception network.

Intuitively, the score increases with the confidence in the classifier prediction (low entropy score) of each image and with the diversity of all images (high overall classification entropy). In this paper, we exploit the shape and texture class information from our two datasets to train two classifiers on top of Resnet-18 [15] features, leading to the *shape score* and *texture score*. Training details of these classifiers are provided in the supplementary material.

AM scores We also use the AM score proposed in [34]. It improves over the inception score by taking into consideration the distribution of the training samples $\bar{x}$ as seen by the classifier C, which we denote $\bar{C}^{train} = \mathbb{E}[C(\bar{x})]$. The AM score is calculated as follows:

$$AM_{score}(\{x\}_1^N) = \mathbb{E}[KL(\bar{C}^{train}||C(x)] - KL(\bar{C}^{train}||\mathbb{E}[C(x)])]$$

The first term is maximized when each of the generated samples is far away from the overall training distribution, while the second term is minimized when the overall distribution of the generations is close to that of the training. In accordance with [34], we find that this score is more sensible as it accounts for the training class distribution.

4.1.2 Mean nearest neighbors distance score

To be able to assess the creativity of the different models while making sure that they are not reproducing training samples, we compute the mean distance for each sample to its retrieved k-Nearest Neighbors (NN), with $k = 10$, as the Euclidean distance between the features extracted from a Resnet18 pre-trained on ImageNet[15] by removing its last fully connected layer. These features are of size 512. This score gives an indicator of the similarity to the training data.

Figure 3: Some generations from a DCGAN with holistic creativity loss applied on texture (Model CAN (H) tex).

A high NN distance may either mean that the generated images has some artifacts, in this case, it could be seen as an indicator of failure, or it could mean that the generation is novel and highly creative.

4.2. Human evaluation setup

We perform a human study where we evaluate different sets of generations of interest on a designed set of questions in order to explore the correlations with each of the proposed automatic metrics in choosing best models and ranking sound generations.

As our RTW garment dataset could not be made publicly available, we conducted two independent studies:

1. Our main human evaluation study was performed on Amazon Mechanical Turk (AMT). 800 images selected randomly or using the automatic metric as described in the paper per model were evaluated, each image evaluated by 5 different persons. There were in average 90 participants per model assessment, resulting in average to 45 images assessed per participant. Since the assessment was conducted given the same conditions for all models, we are confident that the comparative study is fair.

2. We conducted another study in our lab were we mixed both generations (500 total images picked randomly from 5 best models) and 300 real down-sampled images from the RTW dataset. We asked if the images were real or generated to about 45 participants who rated 20 images each in average. We obtain 20% of the generations thought to be real, and 21.5% of the original dataset images were considered to be generated.

Figure 4: First column: random generations by the CAN (H) shape model. Four left columns: Retrieved Nearest Neighbors for each sample.

Creating evaluation sets Given a model's generations, we can extract different clusters of images with particular visual properties that we want to associate with realism, overall appreciation and creativity. Based on the shape entropy, texture entropy and mean nearest neighbors distance of each image we can rank the generations and select the ones with (i) high/low shape entropy, (ii) high/low texture entropy, (iii) high/low NN distance to real images.

We also explore random and mixed sets such as *low shape entropy* and *high nearest neighbors distance*. We expect such a set to contain plausible generations since low shape entropy usually correlates with well defined shapes, while high nearest neighbor distance contains unusual designs. Overall, we have 8 different sets that may overlap. We choose to evaluate 100 images for each set.

We present two strategies to select the best image generations. In the first one, we manually select for each setup four saved models after a sufficient number of iterations. Our models produce plausible results after training for 15000 iterations with a batch size of 64 images. The second strategy employs automatic criteria based on the shape score of generations at a given epoch to discard failed iterations of diverged models. The automatic metrics computed from both epoch selection strategies were found to be very similar. Given the selected models, 10000 images are generated from random numbers – or randomly selected masks for the styleGAN model – to produce 8 sets of 100 images each.

Method/Score	shape	tex.	AM sh	AM tx	NN
Dataset	6.25	3.76	20.4	12.6	5.65
GAN	4.70	2.74	13.3	8.92	**14.4**
GAN classif	**5.31**	2.86	**14.8**	**9.68**	13.1
CAN shape	**5.27**	2.77	**14.7**	8.92	13.1
CAN tex	**5.24**	3.01	**14.4**	**9.48**	**13.5**
CAN shTex	5.20	**3.24**	**14.7**	**10.0**	13.1
CAN (H) shape	5.07	2.80	13.6	8.90	13.0
CAN (H) tex	5.14	**3.33**	**14.4**	**9.30**	**13.6**
CAN (H) shTex	4.98	3.04	13.3	**9.52**	13.2
StackGAN classif	5.20	2.51	**15.2**	8.92	*12.7*
Stack CAN (H) shape	5.13	3.2	13.6	9.02	**13.5**
Stack CAN (H) tex	5.09	3.01	**14.4**	8.25	13.3
Stack CAN (H) shTex	4.82	2.99	13.3	8.85	13.4
Stack CAN shape	5.21	2.65	**14.4**	7.94	13.0
Stack CAN tex	**5.22**	2.87	**14.6**	8.63	*12.7*
Stack CAN shTex	5.15	3.30	13.9	9.08	13.4
Style GAN classif	4.95	2.89	13.6	8.43	**13.5**
Style CAN (H) tex	**5.29**	2.78	13.6	8.19	*12.9*
Style CAN tex	5.15	2.60	13.5	**9.31**	13.4

(a) Ready To Wear dataset

Method/Score	shape	tex.	AM sh	AM tx	NN
Dataset	4.47	4.01	9.13	8.54	12.9
GAN	3.01	2.49	6.38	5.31	**18.1**
CAN (H) shape	3.15	**2.90**	6.90	6.29	*17.3*
CAN (H) tex	**3.20**	**2.83**	**7.21**	6.03	17.6
CAN (H) shTex	**3.28**	**2.83**	**7.21**	6.03	17.6
CAN shape	3.11	2.67	7.13	5.88	17.7
CAN tex	3.09	2.64	6.76	6.63	17.9
CAN shTex	3.16	**2.85**	**7.45**	**6.77**	18.0

(b) Attribute bags dataset

Table 2: Quantitative evaluation on the RTW dataset and bag datasets. High scores appear in bold, low NN distance scores in italic.

Human evaluation questions For computer creativity and fashion, human evaluation is the most reliable measure of soundness of an item, even if it is subjective. Each subject is shown images from the 8 selected sets described in the previous section and is asked 6 questions:

- Q1: how do you like this design overall on a scale from 1 to 5?
- Q2/Q3: rate the novelty of shape (Q2) and texture (Q3) from 1 to 5.
- Q4/Q5: rate the complexity of shape (Q4) and texture (Q5) from 1 to 5.
- Q6: Do you think this image was created by a fashion designer or generated by computer? (yes/no)

Each image is annotated by taking the average rating of 5 annotators.

5. Experiments

This section first presents the datasets used in this paper, then some quantitative results followed by our human experiments that allow us to identify the best models.

5.1. Datasets

Unlike similar work focusing on fashion item generation [19, 36], we choose datasets which contain fashion items in uniform background allowing the trained models to learn features useful for creative generation without generating wearers face and the background.

We augment each dataset 5 times by jittering images with random scaling and translations.

The RTW dataset We have at our disposal a set of 4157 images of different Ready To Wear (RTW) items of size 3000×3760. Each piece is displayed on uniform white background. These images are classified into seven clothes categories: jackets, coats, shirts, tops, t-shirts, dresses and pullovers, and 7 texture categories: uniform, tiled, striped, animal skin, dotted, print and graphical pattern.

Attribute discovery dataset We extracted from the Attribute discovery dataset [2] 5783 images of bags, keeping for our training only images with white background. There are seven different handbags categories: shoulder, tote, clutch, backpack, satchel, wristlet, hobo. These images are also classified by texture into the same 7 texture classes as the RTW dataset.

5.2. Automatic evaluation results

We experiment using weights λ_{G_e} of 1 and 5 for the holistic creativity loss. It appeared that the weight 1 worked better for the bags dataset, and 5 for the RTW dataset. We also tried different weights for the CAN loss but they did not have a large influence on the results and was fixed to 1. All models were trained using the default learning rate 0.002 as in [25]. Our different models take about half a day to train on 4 Nvidia P100 GPUs for 256×256 models and almost 2 days for the 512×512 ones. In our study, it was more convenient from a memory and computational resources standpoint to work with 256×256 images but we also provide 512×512 results in Figure 1 to demonstrate the capabilities of our approach.

Table 2 presents shape and texture scores, AM scores (for shape and texture) and average NN distances computed for

each model on 4 selected iterations. We also computed variances of each score across these 4 iterations and provide the numbers in the supplementary material.

Our first observation is that the DCGAN model alone seems to perform worse than all other tested models with the highest NN distance and lower shape and texture scores. The value of the NN distance score may have different meanings. A high value could mean an enhanced creativity of the model, but also a higher failure rate.

For the RTW dataset, the two models having high shape score, AM shape score, AM texture score and NN distances scores are DCGAN with creativity losses models, holistic or not. On the handbags datasets, the models obtaining the best metrics overall are the DCGAN with creativity losses on shape and texture. To show that our models are not reproducing exactly samples from the dataset, we display in Figure 4 results from the model having the lowest NN distance score, with its 4 nearest neighbors. We note that even if uniform bags tend to be similar to training data, complex prints show high differences.

Figure 5: From the mask of a product, StyleGAN architecture generates different styled image for each style noise.

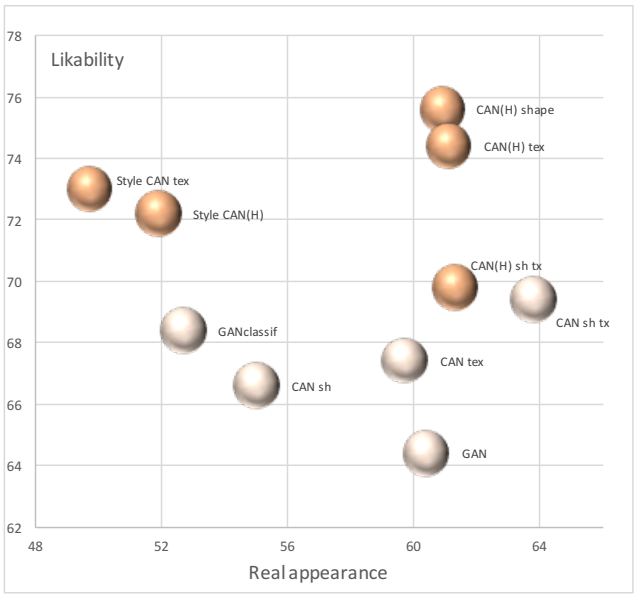

Figure 6: Evaluation of the different models on the RTW dataset by human annotators on two axis: likability and real appearance. Our models are highlighted by darker colors.

Method/Human Method	over-all	shape nov.	shape comp.	tex. nov.	tex. comp.	real fake
DCGAN MCEshape	**3.78**	**3.58**	**3.57**	**3.64**	**3.57**	**60.9**
DCGAN MCEtex	**3.72**	**3.57**	3.52	**3.61**	**3.58**	**61.1**
StyleGAN CANtex	3.65	3.37	3.31	3.44	3.21	49.7
StackGAN	3.62	3.45	3.38	3.43	3.33	51.9
StyleGAN MCEtex	3.61	3.38	3.29	3.50	3.37	53.4
StackGAN MCEtex	3.59	3.36	3.31	3.44	3.28	55.9
StyleGAN	3.59	3.28	3.21	3.27	3.15	47.2
StackGAN CANshape	3.51	3.56	**3.56**	3.58	3.40	50.7
DCGAN MCEshtex	3.49	3.40	3.24	3.40	3.31	**61.3**
StackGAN CANtex	3.48	**3.57**	3.54	3.55	3.50	48.4
DCGAN CANshtex	3.47	3.28	3.18	3.33	3.16	**63.8**
StackGAN MCEshape	3.45	3.27	3.16	3.28	3.12	60.4
StackGAN CANshtex	3.42	3.37	3.32	3.44	3.32	49.5
DCGAN classif	3.42	3.32	3.32	3.37	3.29	52.7
DCGAN CANtex	3.37	3.23	3.12	3.35	3.09	59.7
DCGAN CANshape	3.33	3.28	3.16	3.27	3.12	55.0
StackGAN MCEshtex	3.30	3.25	3.28	3.31	3.27	41.5
DCGAN	3.22	2.95	2.78	3.24	2.83	60.4

Table 3: Human evaluation on the 800 evaluation images from all sets ranked by decreasing overall score (higher is better) on the RTW dataset.

5.3. Human evaluation results

Table 3 presents the average score obtained by each model on each human evaluation question for the RTW dataset.

From this table, we can see that using our creativity loss (CAN (H) shape and CAN (H) tex) performs better than the DCGAN baseline.

While the two proposed models with holistic creativity loss rank the best in the RTW dataset on the overall score, we observe that the preferred images have low nearest neighbor distance which may also be deduced from the correlation Table 4 showing Pearson correlation scores between the 5 automatic metrics and the 6 human evaluation questions. This means that generations which are not close to their nearest neighbors are not always pleasant. It is indeed a challenge to obtain models able to generate novel (high nearest neighbor distance) and at the same time pleasant generations. However, we observe that the models that score better in the high nearest neighbors distance set are clearly the ones with our creativity loss. Figure 6 shows

how well our approaches worked on two axis: likability and real appearance. The most popular methods are obtained by the models employing creativity loss (CAN) and in particular our proposed Holistic loss, as they are perceived as the most likely to be generated by designers, and the most liked overall. We are greatly improving the state-of-the-art here, going from a score of 64 to more than 75 in likabity from classical GANs to our best model with shape creativity. We display images which obtained the best scores for each of the 6 questions in Fig. 8. Our proposed Style GAN and StackGAN models are producing competitive scores compared to the best DCGAN setups with high overall scores.

We may also recover the famous Wundt curve mentioned in the introduction by plotting the likability as a function of the novelty in Figure 7. The novelty was difficult to assess by raters solely from generated images, therefore in this diagram, we give the novelty as the average between shape and texture novelty corresponding to human ratings, and normalized average distance from training images. We note that the least novel model is the GAN with the classification loss, this makes perfect sense as it is trained to generate recognizable shapes and textures. The most novel images, here in term of a large distance to the training images, are the ones generated by the classical GAN. This kind of novelty is however not very much appreciated by the raters. Again, at the top of the curve, our new models based on the holistic loss manage to find a nice compromise between novelty and hedonic value.

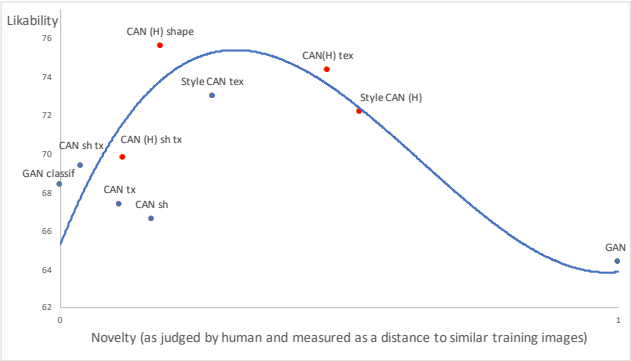

Figure 7: Empirical approximation of the Wundt curve, drawing a relationship between novelty and appreciation. Models employing our proposed Holistic loss appear in red.

Human Auto	over- all	shape nov.	shape comp.	tex. nov.	tex. comp.	real fake
shape	0.3	**0.49**	**0.46**	0.36	0.38	0.06
tex	-0.09	-0.03	-0.05	-0.06	0.08	0.34
AM shape	0.07	0.31	0.31	0.16	0.21	0.06
AM tex	-0.08	-0.16	-0.18	-0.21	-0.17	**0.43**
NN dist	**-0.41**	**-0.71**	**-0.69**	-0.48	**-0.64**	0.23

Table 4: Correlation scores between human evaluation ratings and automatic scores considering all evaluation sets.

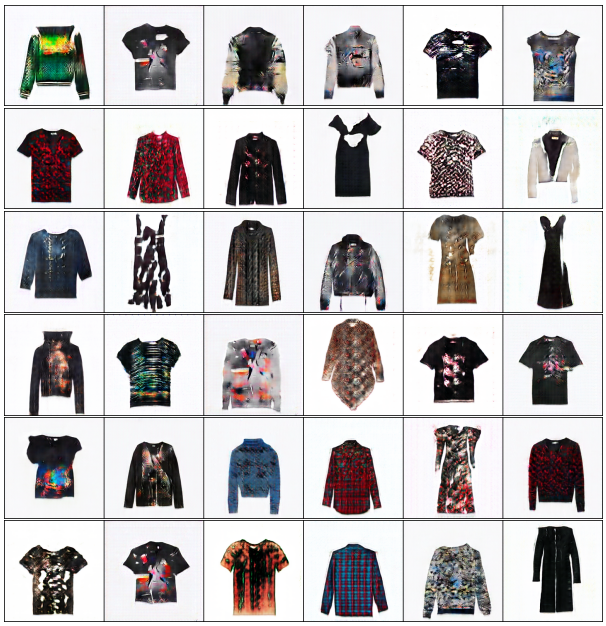

Figure 8: Best generations of RTW items as rated by human annotators (5/5). Each question is in a row. Q1: overall score, Q2: shape novelty, Q3: shape complexity, Q4: texture novelty, Q5: texture complexity, Q6: Realism.

Figure 9: Best generations of handbags as rated by human annotators (5/5). Each question is in a row. Q1: overall score, Q2: shape novelty.

Correlations between human evaluation and automatic scores From Table 4, we see that the automatic metrics that correlate the most with the overall score are the NN dist with -0.41 and the shape score with 0.3. We also observe from this table that the texture scores, and in particular the AM texture score are correlated with the perception of realism of the items. The shape and AM shape scores are correlated to the rates of shape novelty and complexity. We find an important negative correlation (-0.71) between shape novelty and nearest neighbor distance measure which explains how raters find novel generated items which are close to the training dataset. This highlights the importance

of displaying nearest neighbors when rating shape novelty and complexity.

6. Conclusion

We introduced a specific conditioning of convolutional Generative Adversarial Networks (GANs) on texture and shape elements for generating fashion design images. While GANs with such classification loss offer realistic results, they tend to reconstruct the training images. Using a holistic creativity loss, we learn to deviate from a reproduction of the training set. We also propose a novel architecture – our StyleGAN model – conditioned on an input mask, enabling shape control while leaving free the creativity space on the inside of the item. All these contributions lead to the best results according to our human evaluation study. We manage to generate accurately 512×512 images, however we seek for better resolution, which is a fundamental aspect of image quality, in our future work. Finally, while our results show visually pleasing textural creativity, it will be interesting to explore larger families of creativity loss functions, notably in terms of shape, and ensure wearability constraints.

7. Acknowledgements

We are grateful to Alexandre Lebrun for his precious help for assessing the quality of our generations. We also would like to thank Julia Framel, Pauline Luc, Iasonas Kokkinos, Matthijs Douze, Henri Maitre, David Lopez Paz and Neil Zeghidour for interesting discussions. We finally thank everyone involved in the dataset creation.

References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016.

[2] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[3] J.-P. Briot, G. Hadjeres, and F. Pachet. Deep learning techniques for music generation-a survey. *arXiv:1709.01620*, 2017.

[4] P. Date, A. Ganesan, and T. Oates. Fashioning with networks: Neural style transfer to design clothes. In *arXiv:1707.09899*, 2017.

[5] E. L. Denton, S. Chintala, a. szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[6] S. DiPaola and L. Gabora. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines*, 10(2):97–110, 2009.

[7] V. Dumoulin, J. Shlens, M. Kudlur, A. Behboodi, F. Lemic, A. Wolisz, M. Molinaro, C. Hirche, M. Hayashi, E. Bagan, et al. A learned representation for artistic style. *arXiv:1610.07629*, 2016.

[8] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone. Creative adversarial networks. In *ICCC*, 2017.

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[11] L. Grady. Random walks for image segmentation. *Trans. Pattern Anal. Mach. Intell.*, 2006.

[12] D. Ha and D. Eck. A neural representation of sketch drawings. *arXiv:1704.03477*, 2017.

[13] G. Hadjeres and F. Pachet. Deepbach: a steerable model for bach chorales generation. *arXiv:1612.01010*, 2016.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv:1703.06870*, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

[16] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. abs/1611.07004, 2016.

[17] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv 1710.10196*, 2017.

[19] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model of people in clothing. *ICCV*, 2017.

[20] P. Machado and A. Cardoso. Nevar–the assessment of an evolutionary art tool. In *Proc. of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, volume 456, 2000.

[21] P. Machado, J. Romero, and B. Manaris. An iterative approach to stylistic change in evolutionary art.

[22] C. Martindale. *The clockwork muse: The predictability of artistic change.* Basic Books, 1990.

[23] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 2015.

[24] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *arXiv:1610.09585*, 2017.

[25] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015.

[26] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *NIPS*. 2016.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

[28] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game

of go with deep neural networks and tree search. *Nature*, 2016.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017.

[30] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. *ECCV*, 2016.

[31] D. Warde-Farley and Y. Bengio. Improving generative adversarial networks with denoising feature matching. *ICLR*.

[32] W. Xian, P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Texturegan: Controlling deep image synthesis with texture patches. *arXiv:1706.02823*, 2017.

[33] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1612.03242, 2016.

[34] Z. Zhou, W. Zhang, and J. Wang. Inception score, label smoothing, gradient vanishing and -log(d(x)) alternative. *CoRR*, abs/1708.01729, 2017.

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv:1703.10593*, 2017.

[36] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy. Be your own prada: Fashion synthesis with structural coherence. *ICCV*, 2017.

A. Appendix: Network architectures

A.1. Unconditional StackGAN

While the low resolution generator is a classical DCGAN architecture starting from noise only, the high resolution one takes an image as an input. We adapt its architecture as depicted in Figure 10 from the generator described in the fast style transfer model [17] so that the output size is 256x256 given a 64x64 image. It uses convolutions for downsampling and transposed convolutions for upsampling, both with kernel size of 3, stride of 2, padding of 1, with no bias and followed by batchNorm (BN) then ReLU layers. The high resolution generator's architecture is described in Table 5.

Layer	#output channels	kernel size	stride	padding
conv	64	7	1	reflect(3)
conv	128	3	2	1
conv	128	3	2	1
Resnet block $\times 4$	256	-	-	-
convT	256	3	2	1
convT	256	3	2	1
convT	128	3	2	1
convT	64	3	2	1
conv	3	7	1	reflect(3)

Table 5: Detailed architecture of the high resolution (second stage) generator of our proposed unconditional stackGAN architecture. Convolution layers (conv and convT) are followed by Batch normalization (BN) and ReLU, except for the last one, with no BN and Tanh non linearity.

A.2. StyleGAN architecture

The architecture of the styleGAN model combines an input mask image with an input style noise as in the style generator described in the structure style GAN architecture [30]. This consists in upsampling the noise input with multiple transposed convolutions while downsampling the input mask with convolutions. Concatenating the two resulting tensors then performing a series of convolutional layers as shown in the table 6 results in a 256×256 generation.

Branch	layer	#input	# output	ker. size	str.	pad.
Shape	conv	3	64	5	2	2
mask	conv	64	128	5	2	2
branch	conv	128	256	5	2	2
Style	fc	100	1024	4	2	1
noise	convT	64	64	4	2	1
branch	convT	64	64	4	2	1
	convT	64	64	4	2	1
	concat					
	conv	320	256	3	1	1
	conv	256	512	3	2	1
	conv	512	512	3	1	1
	convT	512	256	4	2	1
	convT	256	128	4	2	1
	convT	128	128	4	2	1
	convT	128	64	4	2	1
	convT	64	3	5	1	2

Table 6: Detailed architecture of our StyleGAN model. Convolution layers are followed by Batch normalization (BN) except for the last one. Conv layers are followed by leacky ReLU, ConvT layers by ReLU except the last layer where the non-linearity is a Tanh.

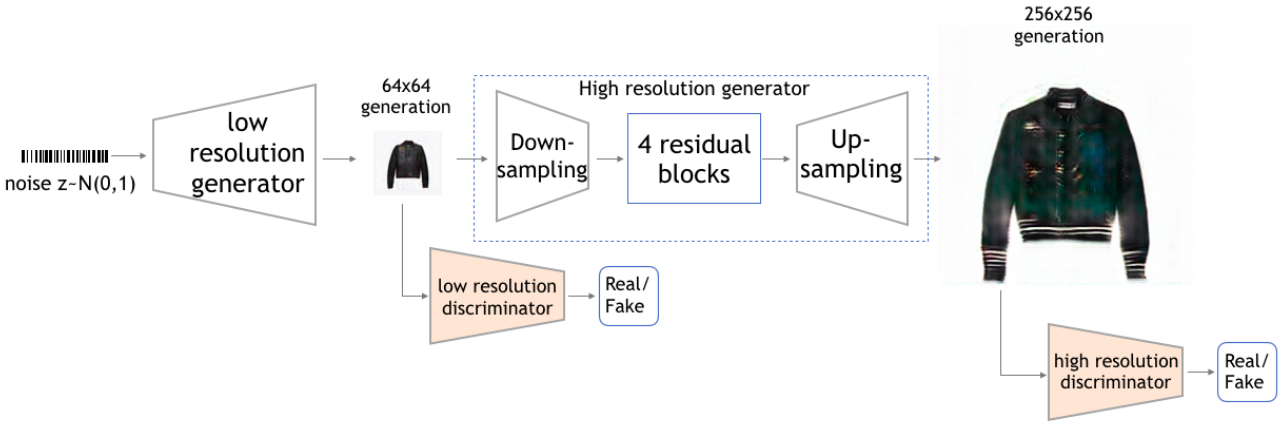

Figure 10: StackGAN architecture adapted without text conditioning.