# ReStGAN: A step towards visually guided shopper experience via text-to-image synthesis

Shiv Surya  Amrith Setlur*  Arijit Biswas  Sumit Negi

Amazon

shisurya@amazon.com, asetlur@cs.cmu.edu, {barijit, suminegi}@amazon.com

## Abstract

*E-commerce companies like Amazon, Alibaba and Flipkart have an extensive catalogue comprising of billions of products. Matching customer search queries to plausible products is challenging due to the size and diversity of the catalogue. These challenges are compounded in apparel due to the semantic complexity and a large variation of fashion styles, product attributes and colours. Providing aids that can help the customer visualise the styles and colours matching their "search queries" will provide customers with necessary intuition about what can be done next. This helps the customer buy a product with the styles, embellishments and colours of their liking. In this work, we propose a Generative Adversarial Network (GAN) for generating images from text streams like customer search queries. Our GAN learns to incrementally generate possible images complementing the fine-grained style, colour of the apparel in the query. We incorporate a novel colour modelling approach enabling the GAN to render a wide spectrum of colours accurately. We compile a dataset from an e-commerce website to train our model. The proposed approach outperforms the baselines on qualitative and quantitative evaluations.*

## 1. Introduction

In large e-commerce companies like Amazon, Alibaba and Flipkart with extensive catalogues, matching customer search queries to plausible products is challenging due to the size and diversity of the catalogue. Customers purchasing products with a personal bias like apparel typically rely on query results to zone in on products matching personal preferences. In apparel, there are a large number of products in a myriad of fashion styles and colour. This means behavioural data is likely to be heavy-tailed. This affects traditional predictive algorithms which rank products in the catalogue based on likelihood of click, purchase or other aggregated customer behavioural data. Helping the customer visualise products with styles and product attributes matching their "search words" will provide customers with necessary intuition about what can be done next. This helps the customer discover and buy products that match personal styles.

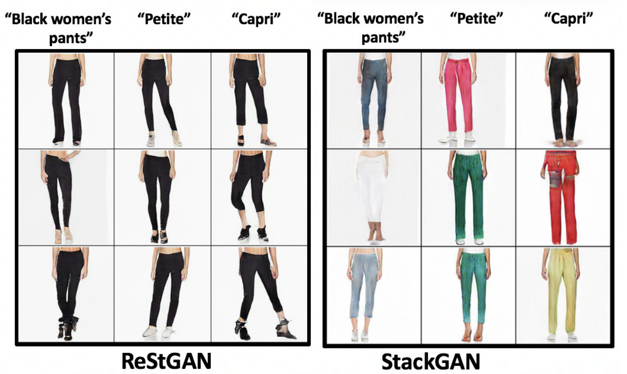

Figure 1. Example prediction by our model, ReStGAN vis-à-vis StackGAN on a text stream of stylistic attributes pertaining to an apparel.

We leverage a Generative Adversarial Network (GAN) [2] to transform stylistic attributes of apparels to images. A GAN is a generative model based on a deep neural network consisting of two components. The former of the two, called the generator $(G)$, transforms random noise to samples mimicking real data. The latter, known as the discriminator $(D)$, inspects the image samples generated by the generator to assert whether they are real or fake. The generator learns to generate samples via feedback from the discriminator. Given a sequence of customer search queries: "Black women's pants" ⇒ "Petite" ⇒ "Capri", the task is to generate a sequence of possible images

[*] Work done at Amazon

matching the queries as they are refined. The natural choice would be to use popular text-to-image GANs to generate an image for each query in the sequence. However text-to-image GANs like StackGAN [24] do not explicitly model sequential data. Fig.1 shows images generated by our model against those generated by StackGAN for the said text sequence. StackGAN fails to maintain consistency of images across the sequence. We overcome this drawback by training a recurrent text-to-image GAN, thus explicitly modelling the sequence. In Fig.1 , our model (ReStGAN) generates images at each step that match the query words at that step while retaining visual attributes from previous generations. This helps the customer to envision possible apparels, which match their evolving queries, and thus guides them to products matching their preference. Our main contributions are:

- The first text-to-image GAN that leverages a recurrent architecture to incrementally synthesise images from a stream of fine-grained textual attributes.
- Novel and effective colour modelling enabling the GAN to render a wide spectrum of colours accurately.
- Quantitative evaluation on a dataset compiled by us from an e-commerce site. ReStGAN achieves a 113% improvement in Inception score-colour, 28% improvement in Inception score-type, 27% improvement in Inception score-gender and 86% reduction in FID score (lower is better) over the traditional StackGAN.

## 2. Related Work

The problem of generating images from textual descriptions has garnered significant traction in the research community. Reliable text-to-image synthesis requires two subproblems to be solved in tandem: compelling image synthesis and a robust natural language representation. Recent strides in image synthesis, building on the family of GAN models [2], have shown evidence of photo-realistic image generation. Several works [14, 16, 1, 7, 12] have incorporated novel optimisation techniques to stabilise the training process and generate striking synthetic images at higher resolutions.

Several extensions to the original GAN formulation have achieved controllable image synthesis by including conditional attributes or class labels in GANs [9, 12, 23, 21]. Text-to-image GANs belong to the family of conditional GANs where the conditioning variable encodes a textual description of the image envisioned. Reed *et al.* [15] use a novel deep architecture coupled with the GAN formulation to generate images from text descriptions. StackGAN [24] improves on Reed *et al.* by using multiple stages to progressively generate high resolution images.

Text-to-image GANs like StackGAN [24], and it's successors [26, 25] are predicated on the assumption that the entire description is present while synthesising the image conditioned on the text. Motivated by the results of text-to-image GANs, we build a model to process text streams and synthesise images incrementally. To the best of our knowledge, our proposed text-to-image GAN bears the distinction that it is the first to leverage Recurrent Neural Networks' (RNN) ability to model sequences and incrementally add fine-grained style details. We enable the training of this novel architecture by integrating it with multiple learning strategies [22, 16, 12]. We demonstrate this model's effectiveness on the case of fine-grained text-to-image synthesis by focussing on an apparel dataset compiled by us.

Prior work in GANs on generation of sequential data [11, 6] has focussed on time-series data generation. Mogren *et al.* use a recurrent generator and discriminator to produce polyphonic music. Hyland *et al.* [6] use a recurrent generator and discriminator to generate medical time series data. They do perform preliminary experiments on generating digits by treating rows in the image of a digit as a sequence while conditioning on the class label of the digit. However, they perform these experiments in a constrained setting with just three digit classes. In contrast, we introduce recurrence in a sophisticated GAN architecture capable of generating photorealistic images from descriptions of apparels. This poses challenges in scale and necessitates the use of multiple training strategies and novel modelling of the conditioning attributes to produce photorealistic images. Our proposed model also produces a distinct high resolution image at each step in the sequence. In contrast, Hyland *et al.* compose a single low resolution image in multiple steps.

## 3. Our Model–Recurrent StackGAN (ReStGAN)

We propose a text-to-image GAN, Recurrent StackGAN(ReStGAN), that leverages Recurrent Neural Networks (RNN) to model sequences of data and generate clothing outfits that envision text descriptions as they appear on the fly. The architecture of ReStGAN is shown in Fig.2. ReStGAN follows a staged approach similar to StackGAN [24] and generates high resolution images through an intermediate low resolution image. The staging makes the generation of high resolution images tractable. ReStGAN has two stages:

**Stage-I ReStGAN:** The Stage-I in ReStGAN comprises of an LSTM that feeds into a convolutional encoder. The LSTM encodes fine-grained text attributes describing the outfit in a hidden representation. The hidden representation corresponding to each fine-grained text input is fed into the upsampling block in Stage-I of ReStGAN, along with noise $z$ and conditioning corresponding to the colour of the item, $c$. The Stage-I generator $G_1$ generates a low-resolution image $I_{lr}$ with the basic contour and colour of the object.

**Stage-II ReStGAN:** Stage-II generator $G_2$ in ReStGAN upsamples the generated image, $I_{lr}$ and adds finer details including texture, stylistic details and colour gradients producing a more realistic high-resolution image $I_{hr}$.

We discuss the architecture of ReStGAN, training techniques, objective modifications and modelling assumptions we incorporate to train ReStGAN in Sections 3.1–3.5.

### 3.1. Stage-I ReStGAN

Stage-I is the first of the stages comprising of an LSTM feeding into a convolutional encoder trained end-to-end. Let $I_r$ be a real image and $y = \{y_1, y_2, y_3, \ldots, y_T\}$ be a sequence of fine-grained text attributes describing $I_r$ drawn from the true data distribution $p_{data}$. Let $z = \{z_1, z_2, .., z_t, .., z_T\}$ be a sequence of noise vectors independently sampled from a given distribution $p_z$ and $\varphi_t$ be the sentence embedding of the given fine-grained attribute $y_t$. $\varphi_t$ is generated by applying a compositional function over word embeddings in the phrase. We use SWEM-concat [18] to generate $\varphi_t$. The generated sentence embedding $\varphi_t$ is fed as an input to the LSTM. For each time step in the forward pass of the LSTM, we get the output hidden state of LSTM, say $h_t$ . We use the hidden state as embedding for text conditioning as it captures the fine-grained attribute at time-step $t$ and historical context. The hidden state $h_t$ is stacked with the colour embedding $c_t$ (see Section 3.4 for details on colour conditioning) at each time-step $t$ to obtain the conditioning $q = \{q_1, q_2, .., q_t, .., q_T\}$. Conditioned on $q$ and random noise variable $z$, Stage-I GAN trains the discriminator $D_1$ and the generator $G_1$ by alternatively maximizing $\mathcal{L}_{D_1}$ in Eq. (1) and minimizing $\mathcal{L}_{G_1}$ in Eq. (2).

$$\mathcal{L}_{D_1} = \mathbb{E}_{(I_r,q)\sim p_{data}}[\sum_{t\in T} \log D_1(I_r, q_t)]+ \mathbb{E}_{z\sim p_z, q\sim p_{data}}[\sum_{t\in T} \log(1 - D_1(G_1(z_t, q_t), q_t))] \tag{1}$$

$$\mathcal{L}_{G_1} = \mathbb{E}_{z\sim p_z, q\sim p_{data}}[\sum_{t\in T} \log(1 - D_1(G_1(z, q_t), q_t))] \tag{2}$$

**Model Architecture** For the generator $G_1$, the hidden state $h_t$ of the LSTM is stacked with random noise vector $z_t$ and colour embedding $c_t$ at each time-step $t$. The resultant $N_g$ dimensional conditioning vector $q_t$ is convolved with a series of of up-sampling blocks to get a $W_1 \times H_1$ image, $I_{lr}$.

For the discriminator $D_1$, the conditioning embedding consisting of the lstm hidden state $h_t$ and the colour embedding $c_t$ are stacked to get an embedding of size $N_d$ dimensions and replicated spatially to form a $M_d \times M_d \times N_d$ tensor. The generated image is encoded by the discriminator encoder and stacked along with the spatially replicated conditioning embedding. The resultant tensor is convolved with a $1\times1$ convolutional layer which projects it onto a lower dimensional space and then a classification layer with a single neuron outputs a decision score classifying it as real or fake.

### 3.2. Stage-II ReStGAN

Low-resolution images generated by Stage-I GAN lack finer details, texture and rich colour gradients that render an image photorealistic. We suitably modify the Stage-II GAN from StackGAN to generate high-resolution images. The Stage-II GAN uses a learnt projection of the hidden state $h_t$ from a fully-connected layer, $\hat{h_t}$, as conditioning along with the colour embedding. Let $\hat{q} = \{\hat{q_1}, \hat{q_2}, \hat{q_3}, \ldots, \hat{q_T}\}$ be the conditioning corresponding to stacked projected embedding $\hat{h_t}$ and colour embedding $c_t$ for all time steps $t$.

Conditioning on the low-resolution result $I_{lr} = G_1(z, q)$ and $\hat{q}$, the discriminator $D$ and generator $G$ in Stage-II GAN are trained by alternatively maximizing $\mathcal{L}_{D_2}$ in Eq. (3) and minimizing $\mathcal{L}_{G_2}$ in Eq. (4). With both $\mathcal{L}_{D_2}$ and $\mathcal{L}_{G_2}$, we use an additional auxiliary classification loss $\mathcal{L}_C$ (Eq. (5)) [12]. $\mathcal{L}_C$ aids in the generation of high resolution images that generate class conditional features which wouldn't be generated if $I_{lr}$ was merely upsampled. We model the auxiliary classification step as a multi-task classification with three independent label spaces ($C$) corresponding to the product type, colour and target gender of the apparel in the image (see Section 3.3 for details).

$$\mathcal{L}_{D_2} = \mathbb{E}_{(I_r,\hat{q})\sim p_{data}}[\sum_{t\in T} \log D_2(I_r, \hat{q}_t)] + \mathbb{E}_{I_{lr}\sim p_{G_1}, \hat{q}\sim p_{data}}[\sum_{t\in T} \log(1 - D_2(G_2(I_{lr}, \hat{q}_t), \hat{q}_t))] + \lambda_1 \mathcal{L}_C \tag{3}$$

$$\mathcal{L}_{G_2} = \mathbb{E}_{I_{lr}\sim p_{G_1}, \hat{q}\sim p_{data}}[\sum_{t\in T} \log(1 - D_2(G_2(I_{lr}, \hat{q}_t), \hat{q}_t))] - \lambda_2 \mathcal{L}_C \tag{4}$$

$$\mathcal{L}_C = \mathbb{E}_{I_r\sim p_{data}}[\sum_{t\in T} \log P(C = c \mid I_r)] + \mathbb{E}_{I_{lr}\sim p_{G_1}, \hat{q}\sim p_{data}}[\sum_{t\in T} \log P(C = c \mid G_2(I_{lr}, \hat{q}_t))] \quad \forall C \tag{5}$$

**Model Architecture** We retain the encoder-decoder network architecture with residual blocks [4] for Stage-II generator from StackGAN [24]. Similar to the previous stage, the projected hidden state $\hat{h_t}$ is stacked along with colour embedding $c_t$ to generate the $N_g$ dimensional conditioning vector $\hat{q}_t$, which is spatially replicated to form a $M_g \times M_g \times N_g$ tensor. Meanwhile, the Stage-I result $I_{lr}$ is encoded using a convolutional encoder block to generate image features. The spatially replicated conditioning is stacked with these image features. The resultant tensor is then feed-forwarded through residual blocks and a decoder to generate a $W_2 \times H_2$ high-resolution image, $I_{hr}$.

The discriminator structure is identical to the Stage-II discriminator in StackGAN with the exception of an auxiliary multi-task classifier. In addition to a real vs fake image classifier, the discriminator has 3 classification layers for tasks pertaining to gender, colour and product type classification . In the form of a regularizer, spectral normalisation [10] is imposed on all layers in the discriminator in Stage-II. In our experiments, we observed this to prevent the generator $G_2$ from collapsing during the initial training epochs.

### 3.3. Tricks for stability and faster convergence

We leverage an auxiliary classifier [12] to stabilise the training of ReStGAN. The auxiliary classification label set $C$ spans gender *(male, female, unisex)*, colour *(see Section 3.4 for details on colour labels)* and product type *(jeans, shorts, pants)* of the outfit being generated. Without the auxiliary classification loss $\mathcal{L}_C$, ReStGAN experienced significant mode collapse.

One-sided label smoothing [16] has been used to encourage the discriminator to estimate soft probabilities and reduce the chances of the discriminator producing extremely confident classifications. While traditionally, only the labels for the real samples undergo smoothing, we smoothen the labels for the fake samples as well. We empirically observed that smoothing the fake labels aided in stabilising losses for the negative pairs in the matching-aware discriminators used to train our GANs (see Section 3.5 for more details on matching-aware discriminator).

#### 3.3.1 Prediction methods for stabilising adversarial training

During training of GANs, training alternates between minimisation and maximisation steps. GAN alternates between updating discriminator $D$ with a stochastic gradient *descent* step, and then updating the Generator, $G$ with a stochastic gradient *ascent* step. When simple/classical SGD updates are used, the steps of this method can be written as in Eq. 6:

$$\begin{aligned} D^{k+1} &= D^k - \alpha_k \mathcal{L}'_D(D^k, G^k) \quad &| \quad &\text{gradient descent in } D \\ G^{k+1} &= G^k + \beta_k \mathcal{L}'_G(D^{k+1}, G^k) \quad &| \quad &\text{gradient ascent in } G \end{aligned} \tag{6}$$

$$\begin{aligned} D^{k+1} &= D^k - \alpha_k \mathcal{L}'_D(D^k, G^k) \quad &| \quad &\text{gradient descent in } D \\ \bar{D}^{k+1} &= D^{k+1} + (D^{k+1} - D^k) \quad &| \quad &\textit{predict} \text{ future value of } D \\ G^{k+1} &= G^k + \beta_k \mathcal{L}'_G(\bar{D}^{k+1}, G^k) \quad &| \quad &\text{gradient ascent in } G \end{aligned} \tag{7}$$

Here, $\{\alpha_k\}$ and $\{\beta_k\}$ are learning rate schedules for the minimisation and maximisation steps, respectively. The stochastic gradients of $\mathcal{L}$ with respect to $D$ and $G$ are denoted by $\mathcal{L}'_D(D, G)$ and $\mathcal{L}'_G(D, G)$ respectively. If either of the steps in Eq. 6 is more powerful than the other, a collapse of the network is observed as the algorithm becomes unstable. Prediction steps [22] mitigate this issue and stabilise the training of adversarial networks by adding a *lookahead* step. An estimate of the position of $D$ in the immediate future assuming current trajectory, $\bar{D}^{k+1}$, is computed. This predicted value of the discriminator is used to obtain $G^{k+1}$. The details are provided in Eq. 7.

We apply prediction steps on both the generator and discriminator networks across both stages. In our experiments with Recurrent GANs, we find that the prediction steps are beneficial in stabilising the training. ReStGAN experienced significant mode collapse without application of prediction steps.

### 3.4. Colour modelling

While prior works in text-to-image GANs including StackGAN [24] feed colour as a part of text conditioning, we find that the embeddings derived from recurrent language models or word embedding spaces like GloVe [13] and Word2Vec [8] do not respect perceptual similarity in the colour space. Sequences S.1–2 in Fig.3 show images generated by a StackGAN model (StackGAN) using text conditioning derived by applying a compositional function on word embeddings of the phrase describing the image. The colour of the fashion item is present in the text phrase. We see that while the stylistic attributes are generated, the colour of the generated samples do not seem to respect the constraint provided by input text conditioning.

To obtain a discriminative representation for colour, we derive coarse clusters of perceptually similar colours that can be mapped to descriptions referencing a particular colour attribute. To generate these clusters for our training data, we use tagged colour attributes (or inferred colour from the text description) from the catalogue (if available). These colour tags/references for products are converted to LAB space using a colour library and clustered using K-Means clustering to generate coarse clusters with similar colours. If a colour tag is absent for a sample, we assign it to a dummy $K + 1$ cluster. With labels generated from this clustering we train a ResNet-50 [3] CNN classifier in a supervised setting. In addition to utilizing the softmax output of this colour classifier as the conditioning for all training examples, we also use it to train the auxiliary classifier in ReStGAN. We find that this mitigates overall noise by correctly classifying examples into clusters which were originally tagged incorrectly in the catalogue.

We find that incorporating colour explicitly as a conditioning improves consistency of colours produced for a given text conditioning. Sequences S.1–2 in Fig.3 compare generated examples for a StackGAN model against a variant of the StackGAN model that explicitly encodes the colour conditioning (StackGAN-C in Fig.3). We see that the consistency of colour across samples and matching of colour to

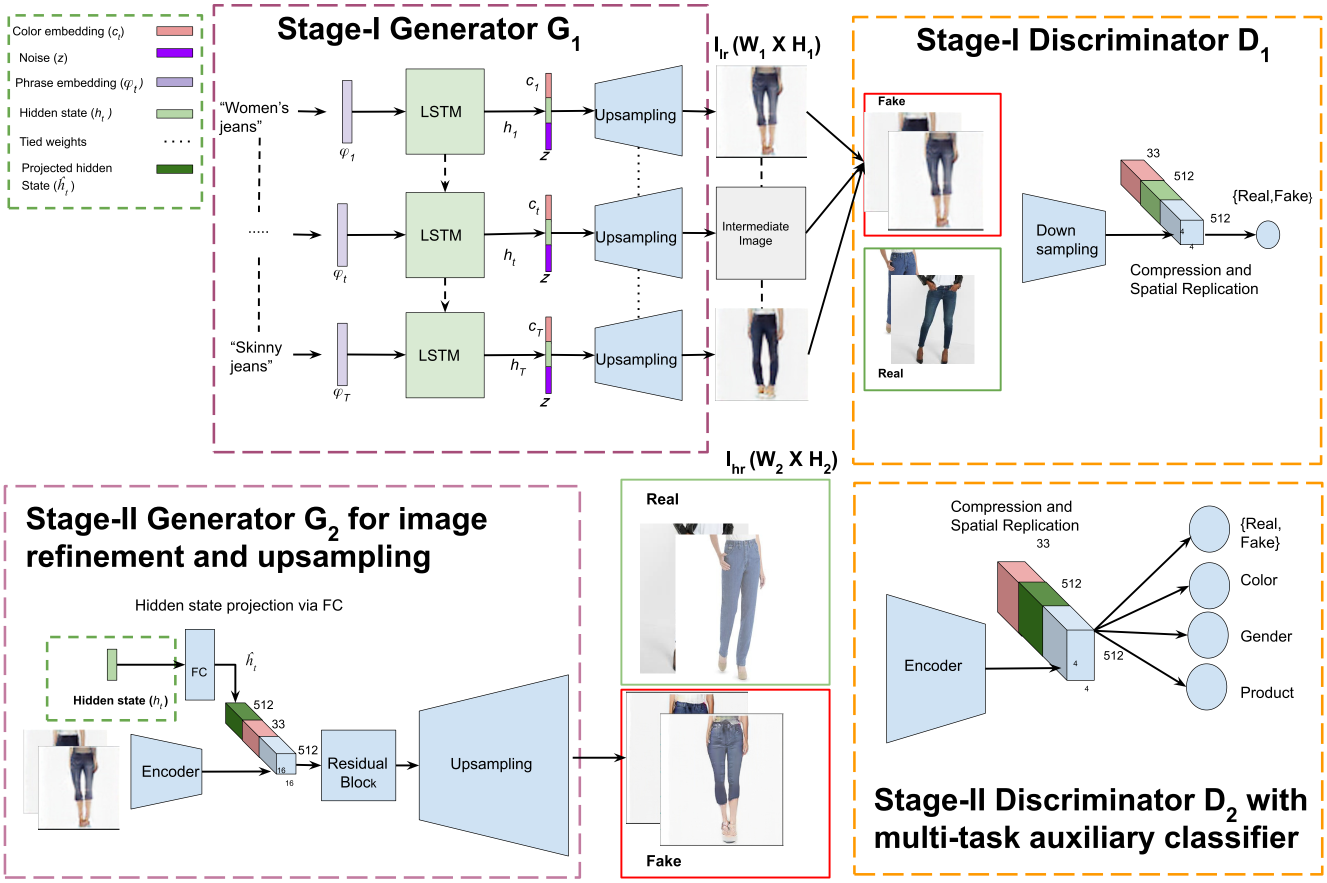

Figure 2. The architecture of the proposed recurrent GAN framework, ReStGAN. In the forward pass, the LSTM encodes the text phrase representation $\varphi_t$, and outputs a hidden representation $h_t$, that encodes each fine-grained text attribute envisioning the outfit. The hidden state corresponding to each fine-grained text input is fed into the Stage-I generator along with noise $z$ and colour conditioning $c$. The Stage-I generator $G_1$ generates a low-resolution image $I_{lr}$ with the basic contour and colour of the object . Conditioned on $I_{lr}$, the Stage-II generator $G_2$ upsamples the generated image and adds finer details including texture, stylistic details and colour gradients producing a more realistic high-resolution image $I_{hr}$.

the textual specification of colour is higher when the proposed colour conditioning is explicitly incorporated.

### 3.5. Training

ADAM solver is used to train $G$ and $D$ across the two stages. For training, we iteratively train recurrent generator $G_1$ and discriminator $D_1$ in Stage-I GAN for 60 epochs with label smoothing. For training $D_2$ and $G_2$, we freeze the LSTM and $G_1$ of Stage-I GAN. The discriminator of Stage-II GAN is trained with the auxiliary multi-task classifier and label smoothing. Prediction steps are applied on both generator and discriminator while training stages I and II. The loss for auxiliary classification tasks for gender, colour and product type classification are scaled inversely by the frequency of classes within each task. All networks were trained with batch size 64 and an initial learning rate of 0.0002. The learning rate is decayed by $\frac{1}{2}$ every 20 epochs. During training of Stage-I/II , input sequences to the LSTM are randomly shuffled and the sequence length is clipped at 6 to ease memory constraints.

The matching-aware discriminator from Reed *et al*. [15] is retained for both stages to explicitly enforce the GAN to learn better alignment between the image and the conditioning. In training a matching-aware discriminator, positive sample pairs *(real images, corresponding conditioning embeddings)* are complemented by negative sample pairs *(real images, misaligned conditioning embeddings)*. The positive and negative pairs are fed as inputs to the discriminator, along with the pairs output by the generator *(generated images, corresponding conditioning embeddings)*. Since we replicate real images (across time-steps) over the input text sequence to the discriminator in ReStGAN, there is a high likelihood that negative sample pairs consisting of real images with misaligned conditioning embeddings may not be truly misaligned. To choose the set of negative sample pairs with the least number of correct alignments, we combina-

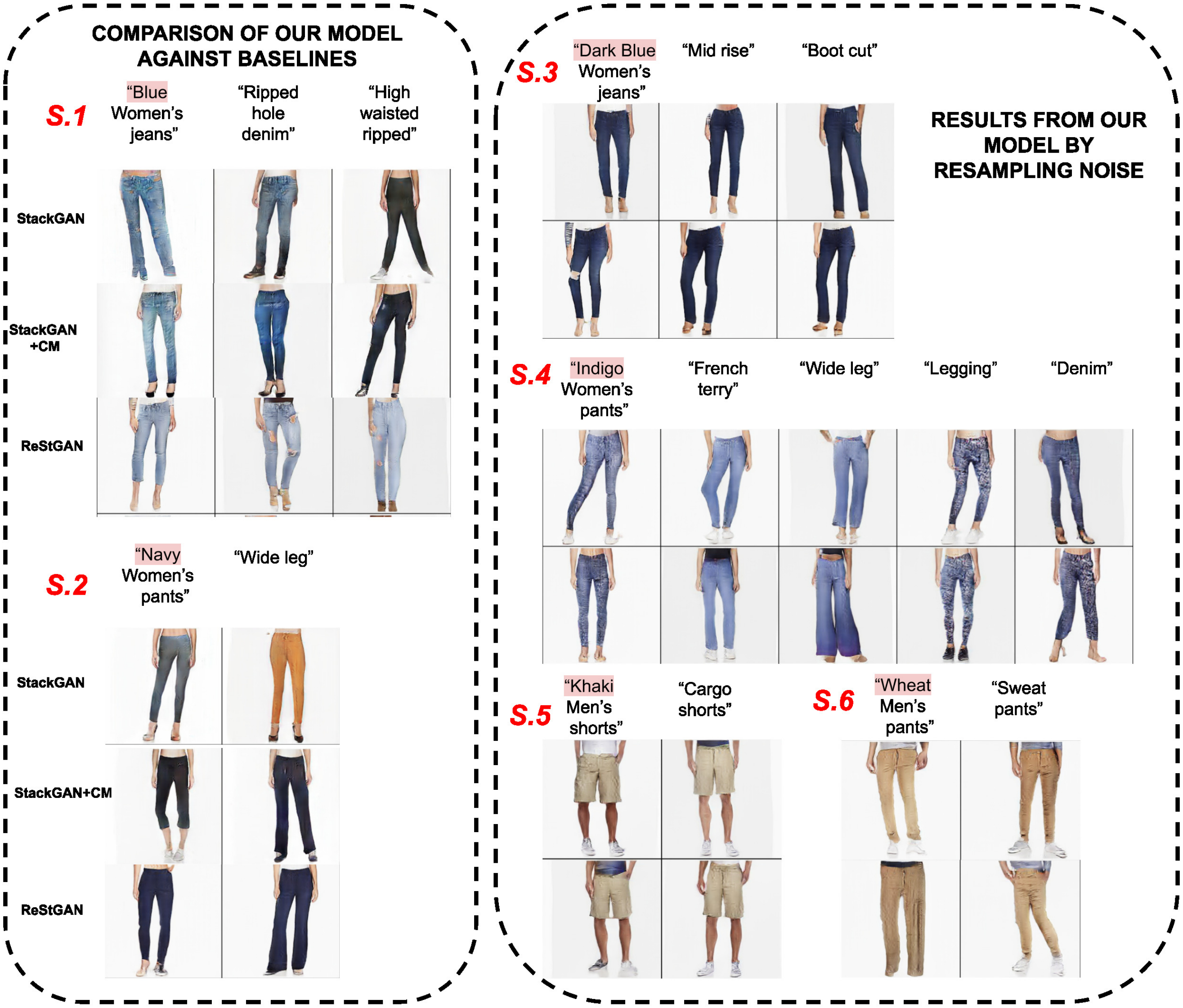

Figure 3. The first block in the image shows the output for ReStGAN and contrasts it against the baseline models StackGAN and StackGAN-C for different text sequences. The second block shows the images generated by ReStGAN for additional text sequences where multiple image sequences are obtained for the same text sequence by jittering the noise $z$.

torially generate multiple sets of negative sample pairs and choose the set with the lowest number of aligned pairs.

The ResNet-50 CNN classifier (see 3.4 for details on classifier) is trained along with auxiliary tasks for gender and product type classification. The auxiliary tasks help in incorporating sample data with no colour labels. For data with absent labels corresponding to one of the tasks, we ignore the loss on the corresponding classification objective. To generate a train and validation split for the multi-label data, we use a multi-label stratification technique [17] implemented in scikit-multilearn package to generate a 80-20 train and validation split.

# 4. Experiments

We compare ReStGAN with baselines including StackGAN and it's variants that ablate the effect of colour modelling, prediction step and auxiliary classifier. More details about baselines and dataset are available in Sections 4.1–4.2.

## 4.1. Baselines

We describe StackGAN and it's variants that ablate the effect of colour modelling, prediction step and auxiliary classifier to enable quantitative evaluation of ReStGAN below. For fairness of evaluation, all StackGAN variants are fed text descriptions incrementally to generate sequences as they do not incorporate explicit sequence modelling.

- **StackGAN:** We use the StackGAN model from which ReStGAN is derived as the primary baseline. The model is trained with all fine-grained text attributes combined into a single textual description. We do not use explicit colour modelling in StackGAN. All loss objectives for training are retained from the original StackGAN model.
- **StackGAN+CM:** This is a variant of StackGAN incorporating the colour modelling. The training hyper-parameters and text conditioning is carried over from StackGAN.
- **StackGAN+PM+CM** This is a variant of StackGAN incorporating the colour modelling and prediction methods applied to both discriminator and generator. The text conditioning is carried over from StackGAN. This model is trained with a higher base learning rate of 0.001 for half the epochs as application of prediction methods enables faster convergence.
- **StackGAN+PM+AC+CM:** This is a variant of StackGAN incorporating the colour modelling, auxiliary classifier and prediction methods applied to both discriminator and generator. All hyper-parameters are carried over from StackGAN+PM+CM.

### 4.2. Dataset

We mainly use an apparel dataset compiled by us from an e-commerce website for training our model. For our experiments, we focus on three product types: pants, jeans and shorts.

**Pre-processing** We apply the following filters on our dataset:

- Hard vote on an ensemble of face detectors with a multi-scale Histogram-of-gradients (HOG) face detector and a CNN based face detector run at multiple scales: faces are hard to model in GANs and we ignore samples which contain faces in our training data.
- Ratio of foreground to background: We use threshold on the foreground to background ratio to remove samples which have close cropped and multi-pack apparel.
- Word filter is applied on textual descriptions for keywords synonymous with baby apparel and printed t-shirts.

Fig.4 shows examples of images that we filter out based on the above preprocessing. From the filtered set, we subsample a training set of 32967 images. The training set is subsampled in such a manner that we get a uniform distribution on the inferred colour. In our final dataset, we have 15372 pants, 12350 shorts and 5245 jeans. All text tokens are generated on this dataset

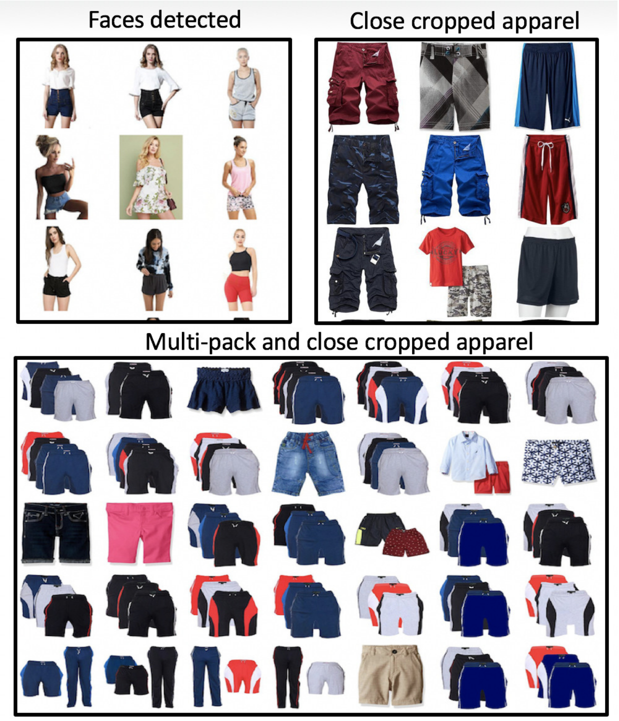

Figure 4. Sample images pruned by pre-processing applied to the initial dataset.

**Text sequence generation:** Since the dataset we compile is not in the form of sequences of customer search queries, we simulate such samples by generating a sequence of fine-grained attributes from an image's description. We synthesize sequences of stylistic attributes from top-k n-grams (1-3 grams) for every apparel type. We filter visually indistinguishable non-stylistic attributes like texture and material. The product type is concatenated with the list of pruned n-grams and this is used as the final sequence of stylistic attributes describing the image of interest. The product type is appended to allow better discriminability among n-grams across product categories which have similar stylistics attributes. For eg: ”Cargo” is a stylistic attribute that occurs in both pants and shorts. Some example text sequences of stylistic attributes can be seen in Fig.3 S.1–6.

### 4.3. Evaluation metrics

We choose two widely accepted metrics for GAN evaluation, namely, inception score [16] and Fréchet Inception Distance (FID) [5] to quantify the performance of ReStGAN against its baselines. The two metrics are formally defined in Eq. 8.

$$
\begin{aligned}
IS_t &= \exp(\mathbb{E}_x[D_{KL}(p(y_t|x)\,||\,p(y_t))]) \\
\text{FID} &= ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})
\end{aligned}
\tag{8}
$$

where $x$ is a sample generated by the model. $y_t$ is the label predicted by the inception model for a task $t$. Task $t$ can be color, gender or product type classification. $\mu_r/\mu_g$ and $\Sigma_r/\Sigma_g$ are the mean and covariance of activations from the Inception classifier [20] corresponding to the real image $x_r$ and the generated image $x_g$. Since our experiments are primarily focussed on a domain specific fine-grained dataset, we fine-tune a trained inception model in a multi-task setting to classify the colour, gender and product type for a given apparel item in our dataset. Computing a score on each of these tasks would capture different facets of image synthesis in an apparel. We compute the inception and FID scores on 32000 samples randomly generated for each model treating each generated image in the sequence to be independent. Lower FID scores and higher inception scores are better. The mean FID and inception scores of 10 runs are reported for every model evaluated.

| Model | IS-Colour | IS-Type | IS-Gender | FID |
|---|---|---|---|---|
| StackGAN | 4.52 | 1.89 | 1.76 | 129.49 |
| StackGAN+CM | 4.25 | 1.91 | 1.75 | 107.84 |
| StackGAN+PM+CM | 4.16 | 1.93 | 1.72 | 99.39 |
| StackGAN+PM+AC+CM | 4.82 | 1.99 | 1.76 | 104.37 |
| ReStGAN(Ours) | **9.65** | **2.43** | **2.23** | **18.71** |

Table 1. Inception scores of ReStGAN and variants of StackGAN with colour modelling(CM), prediction method(PM) and auxiliary classifier(AC). Higher inception scores and lower FID scores are indicative of better image quality.

| Sequence Step(t) | $t = 1$ | $t = 2$ | $t \geq 3$ | $t \geq 4$ | All |
|---|---|---|---|---|---|
| ReStGAN(Ours) | 17.70 | 18.77 | 20.23 | 25.57 | 17.62 |
| Real data | – | – | 1.79 | 9.15 | – |

Table 2. FID scores across time steps for ReStGAN and subsets of real data sampled based on sequence length

### 4.4. Results

Qualitative results for ReStGAN and baselines have been compiled in Fig.3. In S.1, ReStGAN is able to capture intricate stylistic details and embellishments in apparel like “tears in jeans” or the “waist profile” while retaining a consistent colour across the sequence. On the other hand, StackGAN fails to incorporate colour in S.2. Incorporating the colour modelling in StackGAN mitigates this issue. However, both StackGAN and variants incorporating colour modelling fail to add stylistic details incrementally. In S.3–6, we see that ReStGAN generates diverse sequences matching the attributes when we resample noise.

We quantify the performance of ReStGAN against StackGAN (and it’s variants that incorporate colour modelling, auxiliary classification and prediction methods) using the inception & FID scores (Table 1). StackGAN with auxiliary classification has lower FID scores than the corresponding model with prediction methods. We believe this is due to the AC-GAN’s tendency to regress to the modes [19], which would reduce the classification loss at the cost of a reduction in the variety of the generated images. We also see that our model improves upon the inception scores (pertaining to colour, gender and type classification) of the baselines. This is indicative of ReStGAN’s ability to generate diverse images at each time step while retaining semantics of the text conditioning. ReStGAN also gives a significant improvement in FID scores over the different variants of StackGAN.

We compute the Fréchet Inception Distance (FID) for generated samples at each step in the sequence for quantifying ReStGAN’s ability to maintain diversity in samples generated across a sequence (Table 2). For sequence lengths greater than two, we collapse generated samples into buckets of step size greater than three and four. This is done to ensure that sufficient generated examples are present at each sequence step to compute FID statistics. For FID score computations of ReStGAN across sequence steps, we maintain the same real image set. We observe that the FID scores across steps are of the same order as the FID obtained by considering all sequence steps. We observe a nominal increase in FID as the sequence progresses. We attribute this to the increase in specificity of apparel categories in the larger valued sequence steps. To verify that this increase in FID is indeed due to specificity of generations, we also compute FID scores for real examples with sequence lengths greater than three and four against all real examples in Table 2. We observe an analogous increase in FID (indicative of shift in distribution) for the real samples with larger sequence lengths.

Thirty seven additional generations by ReStGAN along with further quantitative analysis is available in the supplementary material.

## 5. Conclusion

We propose ReStGAN for generating images from text streams like customer search queries. It learns to incrementally generate possible images complementing the fine-grained style, colour of the apparel in the query. Additionally, we incorporate a novel colour modelling approach enabling the GAN to render a wide spectrum of colours accurately. We also compile a dataset from a popular e-commerce website’s catalogue to train ReStGAN. The proposed approach outperforms the baselines on qualitative and quantitative evaluations. In future work, we would like to expand ReStGAN’s scope to more vivid apparel types.

## References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *International Conference on Representation Learning*, 2017. 2

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *arXiv prepring arXiv:1506.01497*, 2015. 4

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. 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. 7

[6] S. Hyland, C. Esteban, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *ICLR*, 2018. 2

[7] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *International Conference on Representation Learning*, 2017. 2

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013. 4

[9] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[10] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 4

[11] O. Mogren. C-rnn-gan: A continuous recurrent neural network with adversarial training. In *Constructive Machine Learning Workshop (CML) at NIPS 2016*, page 1, 2016. 2

[12] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 2, 3, 4

[13] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 4

[14] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Representation Learning*, 2016. 2

[15] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *33rd International Conference on Machine Learning*, pages 1060–1069, 2016. 2, 5

[16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 2, 4, 7

[17] K. Sechidis, G. Tsoumakas, and I. Vlahavas. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer, 2011. 6

[18] D. Shen, G. Wang, W. Wang, M. Renqiang Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *ACL*, 2018. 3

[19] R. Shu, H. Bui, and S. Ermon. Ac-gan learns a biased distribution. In *NIPS Workshop on Bayesian Deep Learning*, 2017. 8

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, 2016. 8

[21] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 2

[22] A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein. Stabilizing adversarial nets with prediction methods. *ICLR*, 2018. 2, 4

[23] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016. 2

[24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. 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. 2, 3, 4

[25] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[26] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy. Be your own prada: Fashion synthesis with structural coherence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1680–1688, 2017. 2