DOI: 10.1111/cgf.14938

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

Controllable Garment Image Synthesis Integrated with Frequency Domain Features

Xinru Liang, Haoran Mo, and Chengying Gao [†]

Sun Yat-sen University

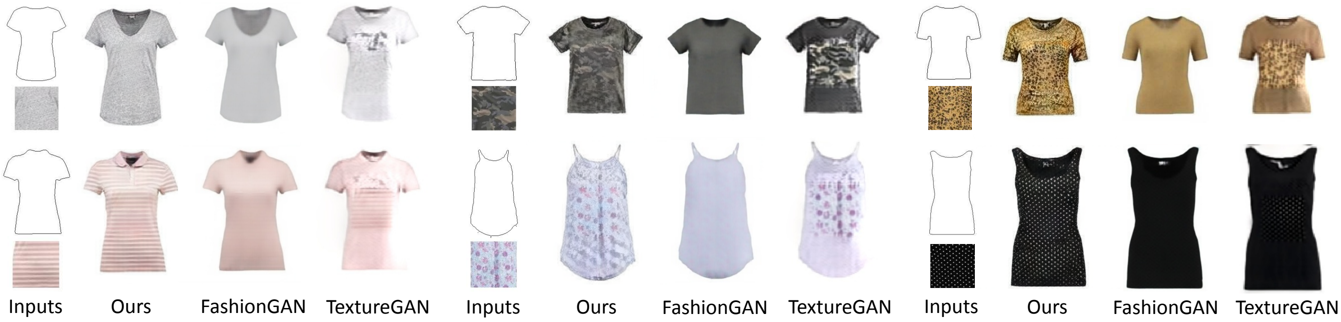


Figure 1: *Comparisons with representative methods [XSA* 18, CLGS18] on garment image synthesis, with a garment sketch and a texture patch as inputs. Existing approaches struggle to produce consistent textures and expand the patterns, while ours works well.*

Abstract

Using sketches and textures to synthesize garment images is able to conveniently display the realistic visual effect in the design phase, which greatly increases the efficiency of fashion design. Existing garment image synthesis methods from a sketch and a texture tend to fail in working on complex textures, especially those with periodic patterns. We propose a controllable garment image synthesis framework that takes as inputs an outline sketch and a texture patch and generates garment images with complicated and diverse texture patterns. To improve the performance of global texture expansion, we exploit the frequency domain features in the generative process, which are from a Fast Fourier Transform (FFT) and able to represent the periodic information of the patterns. We also introduce a perceptual loss in the frequency domain to measure the similarity of two texture pattern patches in terms of their intrinsic periodicity and regularity. Comparisons with existing approaches and sufficient ablation studies demonstrate the effectiveness of our method that is capable of synthesizing impressive garment images with diverse texture patterns while guaranteeing proper texture expansion and pattern consistency.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics;*** *Computer vision;*

1. Introduction

Clothing is always a fundamental part of human life. Nowadays humans' demands for the latest fashions are rapidly growing, which increases the requirement for fast garment design and manufacturing in the garment industry. Garment design is largely limited to manual expert workflows and the huge amount of work dramatically increases the burden of the fashion designers. Therefore, automating some processes in the garment design and manufacturing pipeline is in high demand in the fashion companies and the apparel industry [FCRC05, AD06].

Using a garment sketch and a texture to quickly synthesize the garment image is able to display the realistic visual effect for the designers in the design phase. Unlike the traditional garment design pipeline in which the visual effect of the designed garment is shown after the manufacturing phase, this method greatly in-

[†] Corresponding author: mcsgcy@mail.sysu.edu.cn

creases the design efficiency. The inspiration of the professional designers tends to start with a rough outline (*i.e.,* a sketch), and the fine details are added later. Hence, displaying the visual effect with texture patterns for the outline automatically can provide the designers with a lot of reference information for the subsequent refinement and detail addition. To this end, we propose a *controllable garment image synthesis* method with a sketch and a texture patch as the input elements. Our approach supports outline sketches independent on the fine details inside, which are beneficial to initial outline design and flexible enough for later editing for the experienced designers and even novice users. Several works [XSA*18,CLGS18,LYH*20,SLL20,YZL*22] have focused on this task, but they do not work well with complicated texture patterns. In contrast, our approach produces high-quality garment images with complicated and diverse patterns, as shown in Figure 1.

Textures used in the garment image synthesis often contain repeated patterns, such as stripes, polka dots and plaids. Even for non-repeated patterns such as camouflage and leopard print, they exhibit distinct characteristics or their own regularities. For example, although the spots in the leopard print are scattered, they are still separated from each other and do not merge into a cluster. Expanding the texture patterns during the garment image synthesis process should thus consider the periodicity or the regularity of the patterns, which is highly related to the global context (*i.e.,* long-range dependency) of the synthesized patterns. With current deep learning techniques that are based on convolution operations with shared kernels in a local receptive field as solutions, the capability of modeling long-term information is somewhat reduced. Fast Fourier Transform (FFT) [BM67] is well-suited in this scenario, which transforms spatial images into the frequency domain where globally periodic information in the spatial domain is represented as local one. Therefore, applying the convolutional kernels to the frequency domain features enables a global receptive field for the long-range spatial features and an efficient solution to modeling the pattern periodicity and regularity. We thus propose to integrate the concept of FFT with our generative process to improve the performance of global texture expansion.

Another important aspect of the synthesized garment images is the consistency with the input texture patch in local regions, which is also measured by periodicity or regularity. We observe that two texture patches with the same pattern but not aligned in pixels have relatively similar amplitude images in the frequency domain, as shown in Figure 3. With this characteristic, we propose a perceptual loss in the frequency domain with the amplitude images from two patches cropped from the synthesized and the target image. This loss measures the consistency in terms of the periodicity, and also reflects the quality of the local details.

We evaluate for our framework through qualitative and quantitative comparisons with the existing methods and comprehensive ablation studies of the proposed techniques. These experiments demonstrate that our approach is superior in expanding patterns globally and generating consistent textures with proper regularity. An additional diversity experiment also shows that our method works well with complicated and diverse texture patterns. We also present an application of our approach in garment editing via the sketches, which shows the generalization ability on unseen sketches.

The main contributions of this work are summarized as follows:

- We present a controllable garment image synthesis framework from a garment sketch and a texture patch. The concept of Fast Fourier Transform (FFT) is integrated for better global expansion of diverse texture patterns that the framework works with.
- We propose a perceptual loss in the frequency domain to further improve the ability of capturing the periodicity of the generated texture patterns and preserving the fine-grained details.
- We demonstrate the superior performance of our approach in synthesizing garment images with complicated texture patterns through comprehensive experiments. A derived application of garment editing via sketches is also introduced.

2. Related Work

2.1. Controllable Garment Design

Controllable garment design [XSA*18, CLGS18, LYH*20, YZL*22] aims to synthesize garment images from a garment sketch and a texture patch for quick garment display. The generated texture of the output image should be properly expanded according to the silhouette of the sketch and simultaneously consistent with the input pattern patch, which is still a challenging task. Several methods focus on this task. TextureGAN [XSA*18] proposes a two-stage training strategy, using garment images and sketches in the first stage and texture patches in the second one for fine-tuning. It introduces local texture losses to improve the fine-grained details of generated textures. In most cases, it produces high-quality results in the region where the texture patch is placed, but fails to expand the pattern outside that region. FashionGAN [CLGS18] maps the texture patch into a latent space, which is passed to a BicycleGAN network [ZZP*17b] along with the input sketch for the generation. The latent space allows for the modeling of unseen textures in the inference stage. It can generate good results with texture patches of flat colors or simple patterns such as stripes. But when the pattern is complicated, such as camouflage patterns, leopard prints or floral designs, it relies heavily on the detailed inner contours from the input sketches, which are tedious to obtain. Li et al. [LYH*20] develop an interactive sketching system for fashion images design. To accommodate complicated texture patterns, it also requires users to draw the colorized contours of the patterns, which is inflexible for fashion designers or novice users. Different from the previous methods with a specific sketch as input, ADIN [YZL*22] first uses a random noise to generate a garment sketch, which is then fed to a rendering generator along with a texture patch for garment image synthesis. It is able to generate plausible textures but still fails in complex patterns, exhibiting discontinuity and inconsistency with the input texture. Moreover, the randomness of the generated sketch makes it difficult to precisely control the shape of the generated garment.

Apart from garment sketch, other methods use as input garment segmentation masks [SEB*18, KKL19] as a condition of the garment shape. Compared with garment sketches, they are not flexible enough for the subsequent garment editing.

Several methods [HLBK18, CMG21, HTB*22, CCC*23] work with a similar task of controllable image synthesis according to a sketch image and a reference or style image. While they are not designed for garment images, they suffer from the insufficient capability of texture generation and propagation.

Compared with the methods above, our approach, with highly flexible controls, is able to synthesize high-quality garment images of diverse and complicated texture patterns. To enhance the global expansion of the patterns and capture their periodicity, we integrate the frequency domain features into our framework through FFT and introduce a frequency perceptual loss to boost the performance.

2.2. Exemplar-based Texture Synthesis

Exemplar-based texture synthesis aims at generating a larger texture image from a reference texture patch [ZZB*18]. Existing methods can be divided into two classes: non-universal and universal texture synthesis.

Non-universal texture synthesis algorithms [GEB15, HVCB21] perform a new execution for each single texture pattern. That is, with deep learning technology, a new tuning of the neural network is required for each pattern, making it infeasible to apply to various textures in the scenario of garment synthesis.

Universal texture synthesis methods [MLD*20, GDNR22] train generative models on texture datasets. Once the training is finished, the trained models can be applied to arbitrary textures. This line of methods essentially focuses on expanding a texture patch into a larger texture image, which is similar to our garment synthesis task with texture expansion alike. However, those methods cannot be directly applied to our task due to some noticeable differences. First, our task needs a sketch as an additional constraint of the propagation boundary for the given texture. Second, the given texture should be warped and its luminance should be changed to simulate folds and shadows on the garment.

2.3. Fourier Transform-based Image Synthesis

2D Fourier Transform converts the spatial images into the frequency domain ones, which reflect the low-frequency and high-frequency information of the images. The low-frequency information represents the spatial regions where intensity changes smoothly, such as a large area with a flat color. In contrast, the high-frequency one represents the regions with intensity changing rapidly, such as edge contours, textures and fine details. Recently, Fourier Transform has been proven to benefit the quality of texture generation in image generation tasks, including image inpainting and texture synthesis [BJV17, MLD*20, ZLL*22, JZYS23].

Image inpainting aims to fill up the missing parts in a given image with holes. One of its challenges is to generate the missing parts with repeating patterns. Recently, Fast Fourier Convolution (FFC) [CJM20] is used to synthesize the periodic patterns in the images. LaMa [SLM*22] proposes an inpainting network containing Fast Fourier Convolution Residual Blocks (FaF-Res) based on FFC. CMGAN [ZLL*22] introduces an architecture consisting of an FFC-based encoder and a cascaded decoder. Jain et al. [JZYS23] use FaF-Res and further propose a Fast Fourier Synthesis Module. Due to the impressive results of these works and the fact that garment textures are usually in a repeating mode, we follow these works and adopt Fast Fourier Transform (FFT) in our garment image synthesis framework to improve the ability of expanding the input textures globally.

In the texture synthesis task, several works use frequency domain image features to enforce constraints when optimizing the synthesis algorithm. Liu et al. [LGX16] propose to incorporate Fourier spectrum constraints into the convolutional neural network (CNN) approach, in order to synthesize textures with large scale regularity. Gonthier et al. [GGL22] combine constraints on statistical features (*i.e.,* Gram matrices) and power spectrum of the image to enable long-range dependency when synthesizing high-resolution textures. With the constraints in the frequency domain, these methods gain performance boosts in reproducing the periodic details of the complicated texture patterns. In our work, we propose a perceptual loss in the frequency domain as a guidance to help capturing the periodicity of the textures in the generated garment images. This loss is also able to preserve the fine-grained details.

3. Method

3.1. Overview

We propose a controllable garment image synthesis framework to produce garment images with diverse textures. It takes as input a garment sketch image and a small texture patch image, and is built upon a conditional generative adversarial network (GAN) [GPAM*20], as illustrated in Figure 2-(b). The main challenge of this task is that the input texture pattern should be properly expanded to cover the inner region of the sketch, while still preserving the periodicity and regularity of the pattern. To this end, we take into account Fast Fourier Transform (FFT) [BM67] that is able to represent periodic information in the frequency domain. This concept is integrated into the generator of our framework to improve the performance of global texture expansion. In addition, we leverage the ability of amplitude images from the Fourier Transform in reflecting the regularity of patterns, and propose a perceptual loss in the frequency domain. This loss compares the similarity of two amplitude images of local patches cropped from the generated and target garment images respectively, which is able to further enhance the quality of the generated textures in terms of periodicity and fine-grained details.

3.2. FFT-based Garment Image Synthesis Framework

As shown in Figure 2-(b), our framework consists of two image encoders to encode the input sketch I_{sketch} and the texture image $I_{texture}$, a spatially corresponding feature transfer (SCFT) module [LKL*20] for feature fusion, a Fourier coarse-to-fine (FcF) generator [JZYS23] built upon Fast Fourier convolutional (FFC) layers [CJM20] for garment image synthesis (I_{out}), and a discriminator.

Before training, a data preprocessing stage is required for data collection. With the real garment images (I_{gt}) from the dataset, we first extract the inner masks (I_{mask}) of the images, which are then

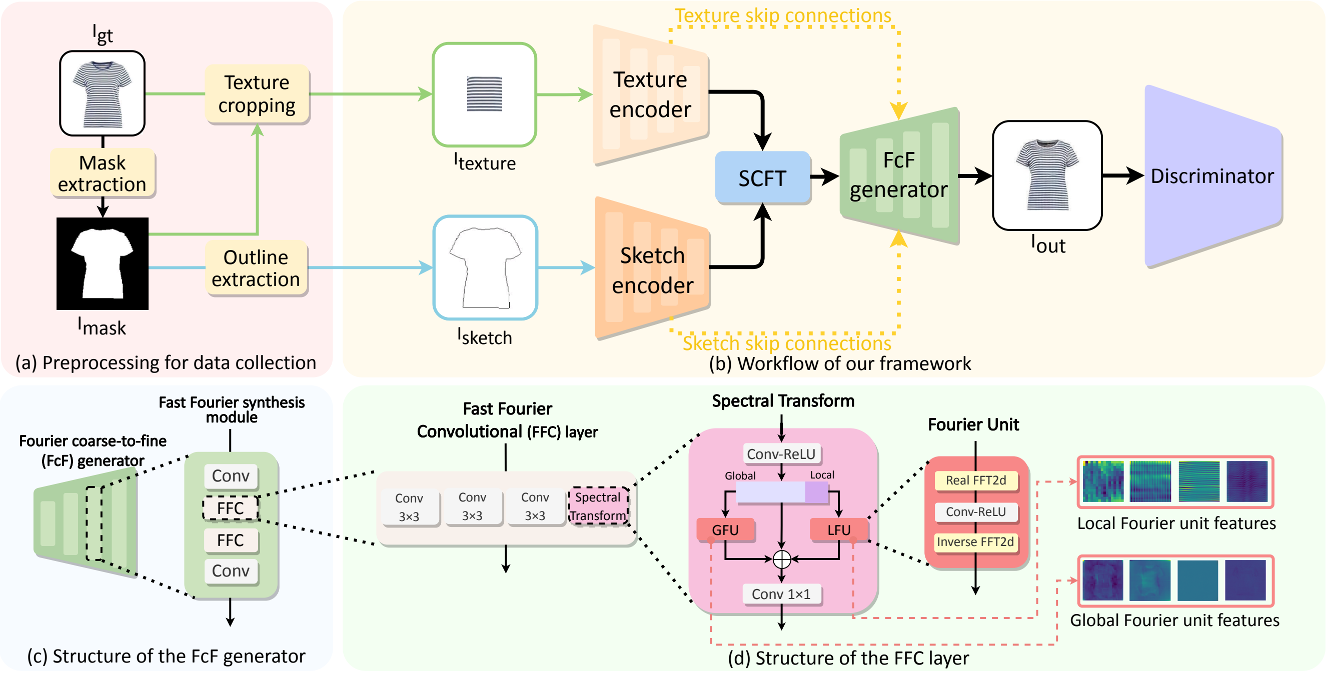


Figure 2: *Overview of our FFT-based garment image synthesis framework (b), which takes as inputs a garment sketch and a texture patch. The training and testing data is collected through a preprocessing stage (a). The FcF generator (c) is integrated with Fast Fourier Convolutional (FFC) layers (d), where a Spectral Transform module and Fourier units are employed to model the global (i.e., the overall structure) and local (i.e., texture details) features.*

processed by an edge detection algorithm to produce the outlines as the sketch images (I_{sketch}). The sketch images in our dataset do not contain any interior contours. Next, a square patch of a random size is randomly cropped from the foreground region of each real garment image indicated by the inner masks. We place the texture patch on the center of a blank image with the same size as the garment sketch, which forms the input texture image ($I_{texture}$). This process helps to accommodate texture patches of arbitrary sizes.

Encoders and Feature Fusion. The two encoders that respectively extract the shape features from the garment sketch image and style features from the texture image share the same architecture and are built upon convolutional layers and a fully-connected output layer. The output feature vectors of the two encoders are then passed to the feature fusion module named SCFT (spatially corresponding feature transfer), an attention-based module proposed by Lee et al. [LKL*20].

SCFT uses the intermediate outputs and the final outputs of the encoder in their original implementation, while we found using the final ones only is sufficient in our task. Specifically, given the encoded texture feature vector $f_{texture} \in \mathbb{R}^{d_f \times 1}$ and the sketch feature vector $f_{sketch} \in \mathbb{R}^{d_f \times 1}$ where d_f is the dimension of the extracted feature vectors, the self-attention mechanism in SCFT uses three learnable matrices $W_q \in \mathbb{R}^{d_f \times d_f}$, $W_k \in \mathbb{R}^{d_f \times d_f}$ and $W_v \in \mathbb{R}^{d_f \times d_f}$ to project the f_{sketch} and $f_{texture}$ into a query $Q \in \mathbb{R}^{d_f \times 1}$, a key $K \in \mathbb{R}^{d_f \times 1}$ and a value $V \in \mathbb{R}^{d_f \times 1}$, respectively (*i.e.*, $Q = W_q f_{sketch}$, $K = W_k f_{texture}$, $V = W_v f_{texture}$). Then, the attention A is calculated as:

$$A = softmax(\frac{QK^T}{\sqrt{d_f}}). \tag{1}$$

The context features are then calculated as:

$$f_c = VA^T. \tag{2}$$

Finally, the context features are added with the sketch features to form the fused feature vector $f \in \mathbb{R}^{d_f \times 1}$:

$$f = f_{sketch} + f_c. \tag{3}$$

FcF Generator. With the fused feature vector from the SCFT module as input and the intermediate features from the encoders as skip connections, our generator aims to synthesize a realistic garment image. The skip connections propagate more information from texture and sketch encoders to the decoder, especially the low-level one such as contour information for ensuring shape similarity and pattern features for recovering fine details. To improve the performance of global expansion of the reference texture patterns, we incorporate our GAN-based framework with the idea of Fast Fourier Transform (FFT) that represents globally periodic information from the spatial domain as local one in the frequency domain. To this end, we adopt a Fourier coarse-to-fine (FcF) generator [JZYS23] originally designed for image inpainting, which

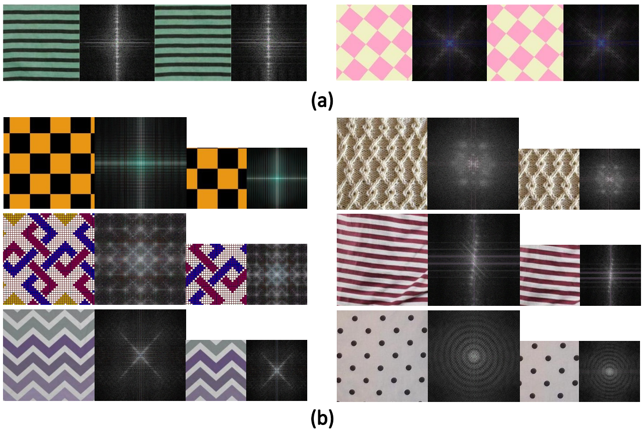


Figure 3: *The FFT amplitude images of texture images with the same pattern, for patches with either the same size (a) or different sizes (b). Although the two patches of each group have different appearances to some extent, their amplitude images in frequency domain are still similar due to the shared pattern periodicity.*

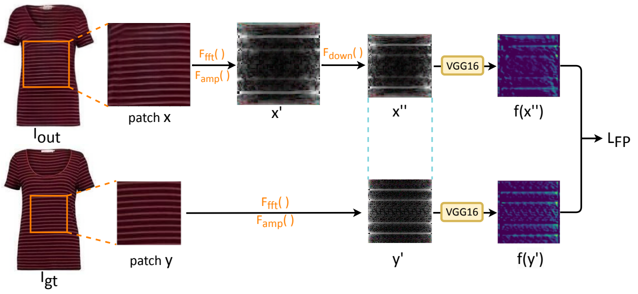


Figure 4: *Workflow of the calculation of the proposed frequency perceptual loss based on a pre-trained VGG-16 model [SZ15].*

combines Fast Fourier convolutional (FFC) layer [CJM20] and a co-modulated StyleGAN2 network [ZCS*21] to control the regularity of the inpainted textures. Fast Fourier convolutional (FFC) layer [CJM20] is proven to be effective in identifying and synthesizing repeated texture patterns.

The FcF generator takes as inputs the fused feature vector f and an additional noise vector z_w converted from a random noise z via a mapping network as in [ZCS*21]. As shown in Figure 2-(c), the FcF generator is built with several synthesis modules from the StyleGAN2 in the starting layers and its proposed Fast Fourier synthesis modules subsequently. The Fast Fourier synthesis module is mainly comprised of Fast Fourier convolutional (FFC) layers [CJM20] (Figure 2-(d)), in which vanilla convolutions with local kernels are employed for the spatial features and a Spectral Transform module is introduced to account for the global and long-range context of the textures. The Spectral Transform module uses a Global Fourier Unit (GFU) to learn the global information (*e.g.,* the overall shape of the garment) and a Local Fourier Unit (LFU) for the semi-global one (*e.g.,* the global repeating patterns), as shown on the right side of Figure 2-(d). The GFU and the LFU share the same structure, with a Real FFT2D operation to convert the spatial contents into image frequencies, a convolutional layer in the frequency domain, and an Inverse FFT2D operation.

Discriminator. The discriminator is borrowed from StyleGAN2 [KLA*20], which takes in inputs the real image and the generated garment image and aims at distinguishing between them. We follow the same adversarial training scheme.

3.3. Frequency Perceptual Loss

Besides the global context and expansion of the texture patterns accounted for by the FFT-based synthesis framework, we also focus on the consistency of the local patches in terms of the periodicity and fine-grained details of the generated patterns. A straightforward solution is to crop a patch from the target garment image and another of the same size and in the same position from the generated image, and then measure the similarity of them. While the expanded texture may not be completely aligned with the target image, it is still perceptually feasible for humans. A case is shown on the top left of Figure 3, where the two texture patches center cropped from the target and synthetic garment images respectively exhibit alternative positions of the green and black stripes. While they appear high dissimilarity over a measurement for spatial images, they are reasonable and acceptable to us due to their similar regularity.

As a result, the problem becomes how to measure the similarity of regularity between two patterns. The FFT amplitude images in the frequency domain that are able to represent the periodic information make them well-suited for this scenario. As shown in Figure 3-(a), two patches that have the same texture pattern but are not spatially aligned share rather similar amplitude images. Moreover, we also notice that even for two patches of different sizes but with the same patterns (Figure 3-(b)), their FFT amplitude images are still spatially aligned after rescaling, which is also observed by Mardani et al. [MLD*20]. According to this observation, we can measure the similarity in terms of the regularity and periodicity by cropping patches from the target and the synthetic garment images, and the croppings can be from different positions and have different sizes. Afterwards, we convert the two patches into their corresponding amplitude images with FFT, and then define a loss to measure the distance between the amplitude images.

Perceptual Loss on the Amplitude Images. The periodicity of patterns reflects the similarity of two texture patches in human perception, even though they have different appearances, as illustrated in Figure 3-(a). This inspires us to use amplitude images storing the periodicity information to account for the texture similarity during training. The amplitude images, different from the natural images with sufficient color gradients, often exhibit dull colors and in a structural appearance. Therefore, we adopt a perceptual loss [MSSG*21] that is able to account for both fine details and overall structure of the shape images.

The calculation of the perceptual loss on the amplitude images is illustrated in Figure 4. We first crop a local texture patch from the synthetic garment image and the target real one, denoted as x and y. As Figure 3-(b) shows when two similar texture patches have different image sizes, their amplitude images are still spatially aligned after rescaling. This allows us to crop patches of arbitrary sizes. A larger patch allows for supervision on a larger area and thus benefits the texture expansion. We thus make x larger than y so as to capture the periodicity of patterns in a larger area of the synthetic image. x and y are then transformed into the frequency domain via FFT, from which we obtain the amplitude images as follow:

$$F_{amp}(I) = log(\sqrt{I_{real}{}^2 + I_{imaginary}{}^2} + 1), \tag{4}$$

where I_{real} and $I_{imaginary}$ denote the real part and the imaginary part of result I after the FFT operation $F_{fft}(\cdot)$. With Eq.(4), we have the amplitude images for the texture patches $x' = F_{amp}(F_{fft}(x))$ and $y' = F_{amp}(F_{fft}(y))$. Then, we downsample the x' to match the size of y', denoted as $x'' = F_{down}(x')$. Afterwards, we use a VGG-16 model [SZ15] trained on ImageNet dataset to extract features for x'' and y' from the intermediate layers. Although trained with natural images, it is found to work on amplitude images. Finally, we calculate the L1 loss between the features from the same intermediate layer. The frequency perceptual loss L_{fp} is defined as follow:

$$L_{fp}(x,y) = \sum_{l \in L_c} \sum_{i=1}^{N_l} \left\| f_i^l(F_{down}(x')) - f_i^l(y') \right\|_1, \tag{5}$$

where $x' = F_{amp}(F_{fft}(x))$ and $y' = F_{amp}(F_{fft}(y))$. L_c is the set of selected intermediate layers of VGG-16, and N_l the number of channels of layer l. $f_l^i(\cdot)$ denotes the feature of the i^{th} channel of layer l.

Cropping Region Selection. We crop patches randomly from the synthetic and the target garment images, while ensuring that the cropped patches should be able to represent the texture pattern of the garment images as much as possible. Thus, we should avoid cropping the background and regions with garment components such as neckline or cuffs.

Such a region is easy to find with our dataset where images have blank backgrounds and garment masks indicating the inner region (foreground) are readily computed. Then, we choose a sub-region on the mask as the cropping region, in which undesired garment components should not be included. The selection is done according to the inherent layout of garments. Specifically, as shown in Figure 5, we calculate the maximum internal rectangle of the mask region, and then drag down the upper bound of the rectangle by 40 pixels (the image size is 256 × 256). Garment components are probably outside such an area. Note that for images with non-pure backgrounds, advanced segmentation techniques such as Segment Anything [KMR*23] could be adopted to extract the foreground masks for the cropping region selection.

3.4. Training

Our GAN-based framework with a generator G and a discriminator D adopts an adversarial training. When training the generator G, besides the adversarial loss $L_{adv}(G)$, we also use a supervision loss L_{sup}, a high receptive field perceptual loss [SLM*22] L_{hp} and our proposed frequency perceptual loss L_{fp} (Eq.(5)). For the discriminator D, the adversarial loss $L_{adv}(D)$ as well as a regularization term L_{reg} are adopted.

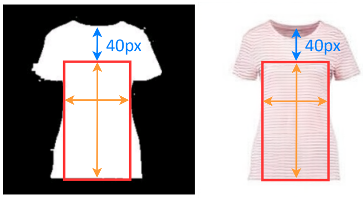


Figure 5: *Selection of the cropping region for the texture patches, which is within the red rectangle.*

Adversarial Loss. We adopt the losses from StyleGAN2 [KLA*20]:

$$L_{adv}(G) = -\mathbb{E}_{I_{sketch}, I_{texture} \sim P_{data}(I_{sketch}, I_{texture})}[D(G(I_{sketch}, I_{texture}))], \tag{6}$$

$$\begin{aligned} L_{adv}(D) =& \mathbb{E}_{I_{sketch}, I_{texture} \sim P_{data}(I_{sketch}, I_{texture})}[D(G(I_{sketch}, I_{texture}))] \\ & - \mathbb{E}_{I_{gt} \sim P_{data}(I_{gt})}[D(I_{gt})]. \end{aligned} \tag{7}$$

Supervision Loss. L1 distance is calculated between the ground truth and the generated image:

$$L_{sup} = \| I_{out} - I_{gt} \|_1 . \tag{8}$$

High Receptive Field Perceptual Loss. This loss [SLM*22] uses a high receptive field-based model to extract the features of the ground truth and the generated image. Following Jain et al. [JZYS23], we utilize the ResNet50 model [HZRS16] pre-trained on ADE20K dataset [ZZP*17a, ZZP*18] for the semantic segmentation task. The loss is formulated as:

$$L_{hp} = \sum_{l \in L_c} \sum_{i=1}^{N_l} \left\| \varphi_i^l(I_{out}) - \varphi_i^l(I_{gt}) \right\|_2, \tag{9}$$

where $l \in L_c$ denotes the intermediate layers we select from the pre-trained ResNet50 model, N_l the number of channels of layer l. $\varphi_i^l(\cdot)$ is the features of the i^{th} channel in layer l.

Regularization Term for Discriminator. Following [KLA*20], we use a regularization when training the discriminator as a gradient penalty term to prevent gradient explosion. This term penalizes the gradients of the discriminator by constraining L2 normalization over them:

$$L_{reg} = \mathbb{E}_{I_{gt} \sim P_{data}} \left[\| \triangledown D(I_{gt}) \|_2 \right] \tag{10}$$

In summary, the total losses for the generator G and the discriminator D are defined as:

$$L_G = L_{adv}(G) + \lambda_{sup} L_{sup} + \lambda_{hp} L_{hp} + \lambda_{fp} L_{fp}, \tag{11}$$

$$L_D = L_{adv}(D) + \lambda_{reg} L_{reg}, \tag{12}$$

where λ_{sup}, λ_{hp}, λ_{fp} and λ_{reg} are scalars.

4. Experiments

4.1. Dataset and Evaluation Metrics

Dataset. We build our own dataset upon the data used in FashionGAN [CLGS18], which contains about 19,000 garment images and their corresponding sketch images with inner contours. We aim at a flexible input sketch that can be easily drawn and edited, so we produce our own sketch images instead of using those in FashionGAN. As illustrated in Figure 2-(a), we first produce the garment masks from the garment images and then extract their outlines as sketches. It is straightforward to extract the garment masks by color-based thresholding, but finding a proper threshold for all the data is practically impossible. We then turn to a learned salient object detection method named BASNet [QZH*19], which is pre-trained on a garment image dataset [WLL*15]. We found it works well in mask extraction on the garment images from FashionGAN. Afterwards, the garment sketches are extracted from the masks via the Canny edge detection algorithm. The masks also serve as the foreground region of the garments, in which the texture patches are randomly cropped and placed on a blank image to form the input texture images. The garment sketches, texture images and the ground truth ones are in a resolution of 256×256. The resolution of the cropped texture patches ranges from 64 to 96. We randomly split 1,000 examples in this dataset as the test set only for evaluation.

Evaluation Metrics. We use Fréchet inception distance (FID) [HRU*17] and learned perceptual image patch similarity (LPIPS) [ZIE*18] as our evaluation metrics. Following Mardani et al. [MLD*20], we also use a cropping-based version of FID (c-FID) and LPIPS (c-LPIPS). For c-FID, we randomly crop 16 texture patches from the output image, and then compute the FID between the input texture and each cropped patch. For c-LPIPS, we do a similar thing except for cropping 8 texture patches. Compared with FID and LPIPS that account for the overall quality, c-FID and c-LPIPS reflect the quality more in local areas or details of the images.

4.2. Implementation Details

Network Details. Our texture encoder and sketch encoder share the same structure as the discriminator used in StyleGAN2 [KLA*20] but without the residual skip connections. Both encoders map the input image into a 1024-dim latent vector. The mapping network for the inputs of the FcF generator has the same settings as [ZCS*21] and consists of a series of fully-connected layers, which converts a 512-dim random noise z to a new noise vector z_w with the same dimension.

Training Details. We train our model on a machine with an NVIDIA GeForce RTX 3090 GPU. We train for 1,120k iterations totally with a batch size of 12. Adam [KB15] is used as the optimizer with an initial learning rate of 1e-4. The loss weights in Eq.(11) and Eq.(12) are set to $\lambda_{sup} = 10, \lambda_{hp} = 5, \lambda_{fp} = 4$ and $\lambda_{reg} = 5$.

4.3. Baseline Methods

We compare with the baseline methods as follows. The hyper-parameters are kept the same as the ones in their original implementations for fair comparisons.

- **FashionGAN** [CLGS18]. It is a garment image synthesis method that is closest to ours, with a garment sketch and a texture patch cropped from a garment image as input. It is re-trained on our own dataset.
- **TextureGAN** [XSA*18]. It is also close to our approach with a garment sketch and a cropped texture patch as input. It originally works with images with a resolution of 128×128, and is found to work worse with a larger resolution such as 256×256 in our dataset. Thus, we downsample our images to 128×128 and re-trained TextureGAN.
- **MUNIT** [HLBK18]. It is an unsupervised method that transfers images from a source domain to a target domain. We treat the garment sketches as the source domain. During training the real garment images are used as the target domain. During the testing stage where only texture patches are provided, we tile them within the inner region of the garment sketch to form the images for the target domain.
- **ReferenceGAN** [LKL*20]. It is a reference-based GAN model for sketch to natural image transfer. We use our garment sketches as its sketch input and our texture images as the reference.
- **SSSIS** [LZSE21]. It is also a reference-based sketch to natural image transfer method. It uses a two-stage generation strategy. In the first stage, it extracts the content and the style features from the two inputs and generates an image with a GAN-based architecture. In the second stage, another GAN-based network is employed to refine the generated image. We only train the first stage since we found the second one degrades the generation quality in our task.
- **DiSS** [CCC*23]. It is a diffusion model-based method that generates a natural image from a stroke-based style image and a sketch, by using the technique of classifier-free diffusion guidance [HS21]. We use our garment sketch images as its sketch input and our texture images as its style input.

4.4. Comparison with Existing Approaches

Qualitative Comparison. Figure 6 shows the results of our approach and the baseline methods, in which we evaluate all the methods with garment sketches of different types (*e.g.,* T-shirt, vest, long-sleeved shirt, etc.), texture patches of different sizes and patterns (*e.g.,* colored fabric, stripe, polka dot, leopard print, camouflage, plaid, etc.). From all the results, the garment images generated by our approach exhibit the best visual quality in terms of color faithfulness, texture consistency, pattern expansion, and fine-grained details. They demonstrate the effectiveness of our proposed FFT-based synthesis framework and the frequency perceptual loss in improving the performance of global texture expansion and pattern periodicity preservation.

Regarding the baseline methods designed for garment image synthesis, FashionGAN is able to generate images with colors largely consistent with the input textures, but fails to reproduce most textures except for the stripe. For those complicated patterns, it tends to produce a blurry and average color in the entire area of garment images. This is probably because FashionGAN uses a texture encoder to map the input texture into a low-dimension latent space so that it allows modeling the unseen textures during inference. Such a low-dimension latent space leads to the loss of the

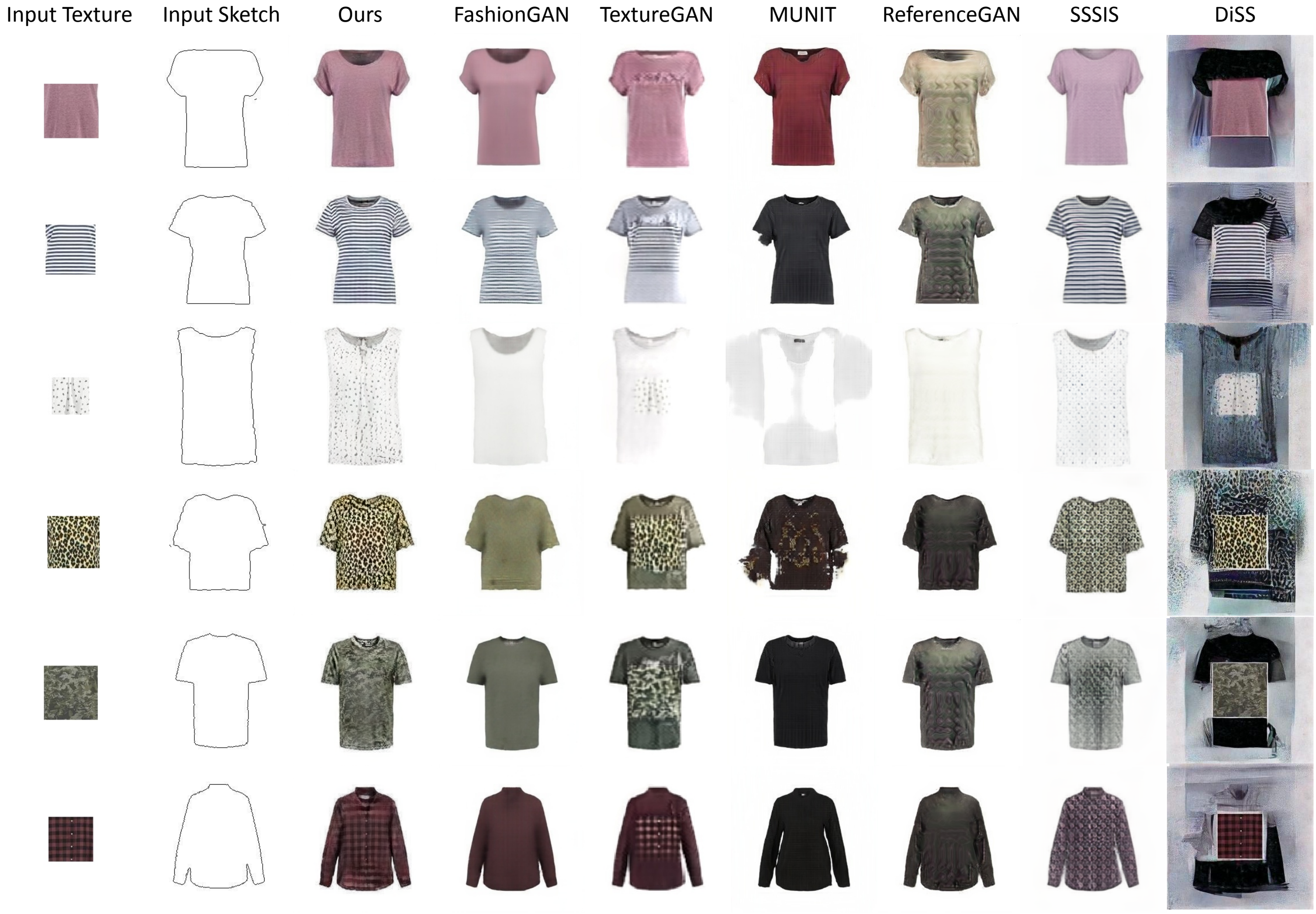


Figure 6: *Qualitative comparison with baseline methods. The texture patch of different sizes and the garment sketch in the first and the second columns are used as the inputs. We show more results in the supplemental material.*

abundant spatial information of the texture patterns. TextureGAN produces a blurry reconstruction of the input textures in the area where they are placed (*i.e.*, the center) and fails to expand them, leaving a flat color outside the area of the texture patch. This indicates even on a low resolution (128px), TextureGAN fails to model the characteristics of the textures for recovering and expansion. When applied to a higher resolution (*e.g.*, 256px), such artifacts (especially the blur) are magnified, leading to an even worse performance.

As for the approaches for sketch to image translation, MUNIT performs much worse in terms of color faithfulness and texture consistency. Moreover, it fails to capture the inner region of the garment sketches. These issues are caused by the unsupervised learning scheme of MUNIT, making it difficult to model the variety of texture patterns in the limited target domain. Similar to MUNIT, ReferenceGAN originally designed for natural images also fails to produce faithful colors and consistent patterns with the input textures, although it is able to capture the shape of the garments. SSSIS generates plausible results with a stripe texture, but seems to produce over-repeated patterns for other complicated textures. We assume this is because the image augmentation operations in its style encoder break the original regularity of the texture pattern, and thus the encoder learns a general pattern with excessive periodicity for all the textures. DiSS effectively reconstructs the input textures in the center area, but cannot expand them properly. Compared with other methods, it lacks the ability to produce a blank background. This is probably because the style input in its original scenario is required to store information of the content (*e.g.,* the overall shape), in order to align the style with the sketch shape during the reversed diffusion process. However, in our task, the input textures are not aligned with the shape of the garment sketches at all.

Quantitative Comparison. We use our test set for the quantitative evaluation. Table 1 shows the results, where our method performs the best on all the metrics. The results of FID and LPIPS indicate that our method has the best overall quality. The c-FID and c-LPIPS confirm that our method is superior to other methods in producing local texture details. These results are consistent with the qualitative ones.

Table 1: *Quantitative comparison with baseline methods.*

	FID↓	LPIPS↓	c-FID↓	c-LPIPS↓
DiSS [CCC*23]	190.969	0.608	124.066	0.569
ReferenceGAN [LKL*20]	70.417	0.236	100.420	0.567
FashionGAN [CLGS18]	54.759	0.161	99.222	0.452
TextureGAN [XSA*18]	46.019	0.303	77.620	0.516
MUNIT [HLBK18]	44.919	0.250	83.576	0.516
SSSIS [LZSE21]	39.121	0.164	70.145	0.445
Ours	**17.146**	**0.126**	**31.796**	**0.371**

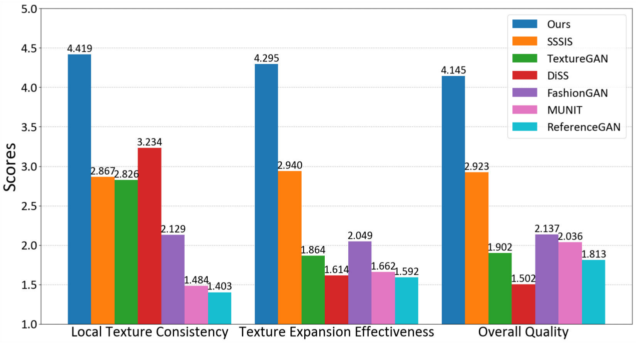


Figure 7: *Results of the user study. The participants gave a score ranging from 1 to 5. Higher scores mean higher preference.*

User Study. We further conduct a user study to compare all the methods. We randomly select 30 examples from our test set and divide them into 3 groups, each of which contains 10 examples. We invite 28 participants for each group (84 participants in total), and ask them to score the synthetic garment images of each method. All participants are from multiple backgrounds and have no prior knowledge of this project. For each result, the participants are asked to score according to three aspects: (1) *Local texture consistency*, which indicates the consistency of pattern and color in local regions with the input texture. (2) *Texture expansion effectiveness*, which means the performance of the model in expanding the texture pattern to the entire inner region of the garment. (3) *Overall quality*, which includes subjective measurements such as realism, light and shadow effect, 3D effect, etc.

The average results are shown in Figure 7. Our approach obtains the best scores in all the aspects, which corroborates the effectiveness of our framework in reconstructing and expanding the texture patterns as well as generating high-quality garment images. Among the baseline methods, SSSIS is overall superior to the others, but is still inferior to ours by a large margin. DiSS works well in guaranteeing local texture consistency, which is in line with the qualitative results where the textures are properly reconstructed in the center area (Figure 6). While its weaknesses in expanding the textures and generating plausible results are also revealed in the user preference.

4.5. Ablation Studies

We conduct three ablation studies to evaluate the importance of each component of our framework.

Table 2: *Quantitative result of ablation studies. Our approach has a FFT-based generator, a dual-branch encoder, and a frequency perceptual loss (L_{fp}).*

	FID↓	LPIPS↓	c-FID↓	c-LPIPS↓
Generator w/o FFT	34.002	0.159	48.054	0.403
Single-branch encoder	21.197	0.141	33.257	0.373
w/o L_{fp}	17.409	0.127	33.155	0.375
Ours	**17.146**	**0.126**	**31.796**	**0.371**

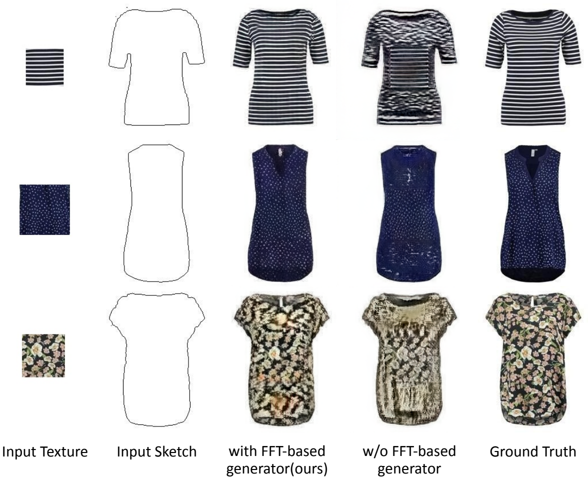
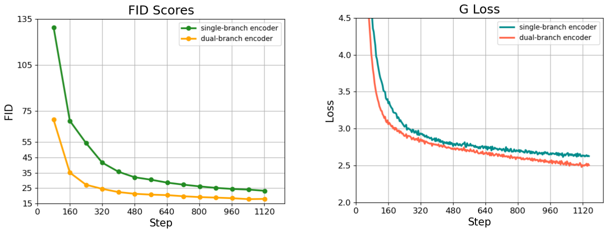


Figure 8: *Comparisons between methods with and without FFT-based generator.*

Effectiveness of FFT-based Generator. The FFT-based generator (FcF generator) is built upon a co-modulated StyleGAN2-based coarse-to-fine generator [ZCS*21], with integration of fast Fourier convolutional layers [CJM20]. Thus, we simply use the co-modulated StyleGAN2 generator without the concept of FFT as the ablation. From the quantitative results in Table 2, we can see that the method without the FFT-based generator has a considerable drop in performance. Figure 8 shows the qualitative differences, where the method without the FFT-based generator is able to reconstruct the input texture in the center area, but works poorly in propagating the patterns globally. In contrast, our approach with a generator integrated with the Fast Fourier Transform-based modules expands the texture pattern well, resulting in high-quality synthetic images.

Encoder Architecture. Our framework employs a dual-branch encoder, *i.e.,* one branch for the input sketch and the other for the input texture image. They encode the two images separately. We also evaluate a single-branch encoder taking as input the concatenation of the sketch and the texture image. In Table 2, we can see that the single-branch method suffers from a certain degree of degradation. We illustrate the training information of the two methods in Figure 9, and we can see that our approach with a dual-branch



(a) FID Score

(b) Total Loss

Figure 9: *Training information between methods with a single-branch and a dual-branch encoder.*

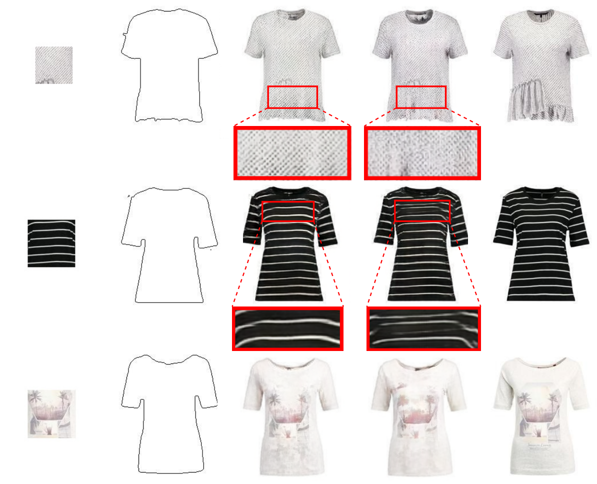


Figure 10: *Comparisons between methods with and without our proposed frequency perceptual loss (L_{fp}).*

encoder converges faster and has a better performance in terms of FID scores.

Frequency Perceptual Loss. We propose a perceptual loss in the frequency domain in order to aid in capturing the periodicity of the texture patterns in the local areas. We evaluate this loss by removing it from our framework. The quantitative results are shown in Table 2, in which the method without this loss is slightly worse than ours with the loss. Although the quantitative difference is insignificant, we observe a noticeable improvement in our qualitative results, as shown in Figure 10. In the first and second rows, our approach with the frequency perceptual loss produces more periodic and more regular patterns that are consistent with the input texture. The quality of the fine-grained details is meanwhile improved. In addition, we also notice that the frequency perceptual loss benefits the reconstruction of the input textures, particularly with a print pattern. As shown in the last row of Figure 10, the print is reconstructed better in our approach.

4.6. Diversity of Texture Patterns

Our controllable garment image synthesis framework is able to work with a variety of texture patterns in a single trained model, unlike those non-universal texture synthesis approaches [GEB15, HVCB21] that require a new tuning of the neural network for each pattern. Figure 11 shows the results from our approach when applying multiple texture patterns to a garment sketch. The textures include common patterns such as plain fabric, camouflage, stripe, polka dot, velour and leopard print, as well as customized ones such as irregular floral print and photo print. For all these texture patterns, our approach generates impressive results.

4.7. More Applications

Controllable Garment Editing. Our controllable garment image synthesis framework works on outline sketches, and thus allows for the visual effect display of not only a well-designed sketch, but also the subsequent editing of a given sketch. This is especially useful for fashion designers in their designing phase, and even novice users when they try out our system. We demonstrate such an application with several edited sketches in Figure 12, where we change the shapes of sleeves (*e.g.,* long sleeves (b) and wide sleeves (e)), collar (*e.g.,* a round collar (c)), shoulder (*e.g.,* a sloping shoulder (h)), and waist (*e.g.,* a shorter waist (f) and an asymmetry waist (i)). For all these edited sketches that are unseen during the training, our framework is able to produce reasonable synthesized results with equivalent quality to the original ones, implying that our approach allows flexible editing and meanwhile has high generalization ability on garment sketches with a wide variety of appearances.

Generalization to Real Garment Sketch. While trained with outline sketches, our approach generalizes to real garment sketch to a certain degree. As shown in Figure 13, they exhibit interior details in collars, cuffs and waists, describing the garment style. Our method recovers the interior details, albeit with results slightly inferior to those on outlines. The performance could be improved if more training data is provided. Our framework mainly works with tops, as we collect them only as our dataset given their varying designs and styles. If more types of garment data (such as bottoms) are available during training, our method could handle more general types of garments theoretically. These could be future works of our approach.

4.8. Resolution Increasing

In the experiments above, we generate the garment images in 256 × 256 to make fair comparisons with existing methods. The resolution can be increased for more realistic results. We adopt the latest technique in Stable Diffusion [RBL*22], where a pre-trained autoencoder is employed for super-resolution of the images generated by the latent diffusion model. Specifically, the 256 × 256 results synthesized by our method are first input to the trained latent diffusion model to denoise for 100 steps, and then converted into 512 × 512 ones by the decoder of the pre-trained autoencoder. The results with increased resolution are shown in 14, where we can see the details are more clear and the entire images are more realistic.

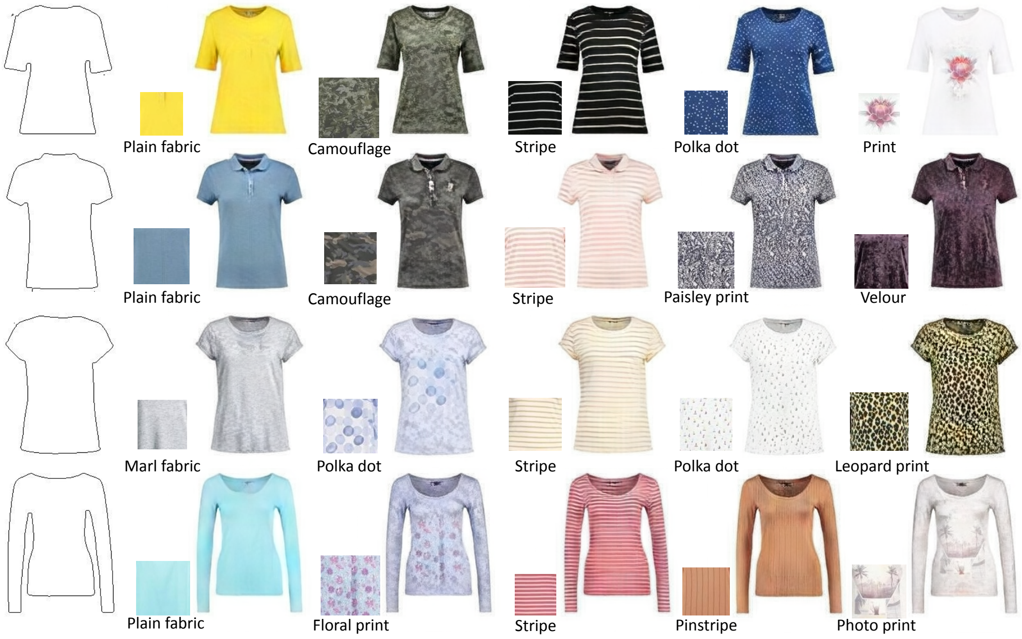


Figure 11: *Diversity results with various textures per garment sketch. We show more results in the supplemental material.*

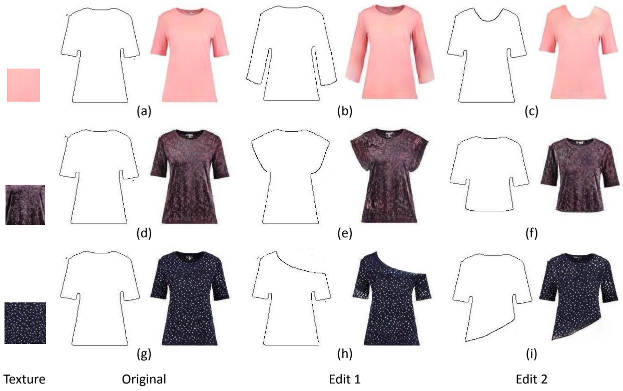


Figure 12: *Editing results on the garment sketches.*

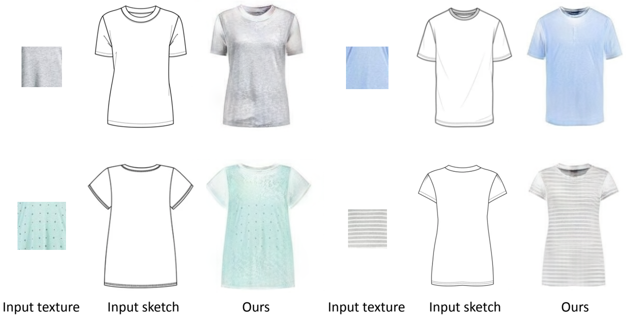


Figure 13: *Generalization to real garment sketches.*

5. Conclusion and Limitations

This paper presents a framework for synthesizing high-quality garment images with a garment sketch and a texture patch as inputs. The generation results display the realistic visual effect of the designed garments, and can increase the design efficiency of the fashion designers. To synthesize garments with properly expanded textures, we integrate our framework with the concept of Fast Fourier Transform (FFT) that enables the framework to model the periodic information of the patterns. To better capture the regularity of the texture patterns, we further propose a frequency perceptual loss based on the characteristics of the amplitude images in the frequency domain. Comprehensive experiments are done to corroborate the effectiveness of our approach.

Although producing impressive results with most complicated texture patterns (Figure 11), our method may still fail in some over-complex and colorful patterns, as shown in Figure 15. In the first row, the pattern contains not only the floral designs, but also different textures such as flat colors and dense dots. This case is especially difficult for our approach to propagate the pattern. In the second row, the stripe pattern contains multiple colors and our result fails to expand this texture. This is probably because this multi-color pattern is different from the common two-color stripe ones with stripes placing alternately. Our method is not yet able to understand the positions of the stripes for each color. To better handle the cases above, structural information of the patterns could be first

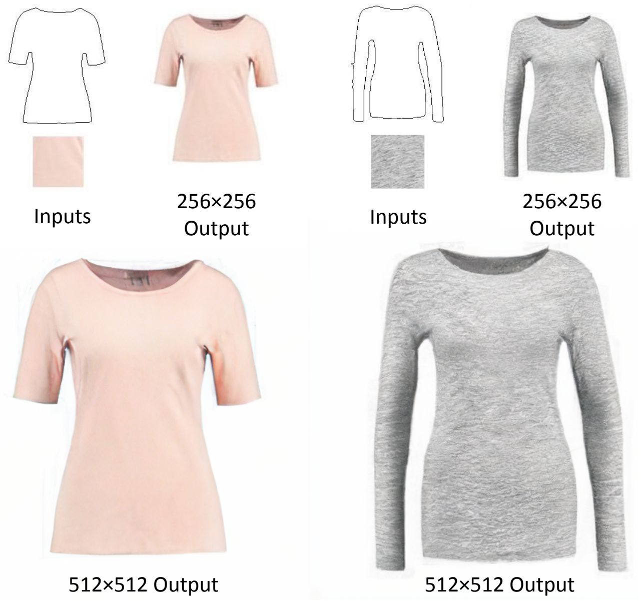


Figure 14: *Results with increased resolution.*

discovered and understood [RGF*20], and then integrated into our framework. This may be a future extension of our work.

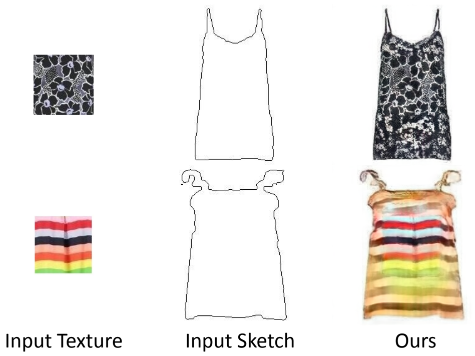


Figure 15: *Limitations of our method in textures with over-complicated patterns and multiple colors.*

Acknowledgments

This work was supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2022A1515011425).

References

[AD06] Ashdown S. P., Dunne L.: A study of automated custom fit: Readiness of the technology for the apparel industry. *Clothing and Textiles Research Journal 24*, 2 (2006), 121–136. 1

[BJV17] Bergmann U., Jetchev N., Vollgraf R.: Learning texture manifolds with the periodic spatial gan. In *International Conference on Machine Learning* (2017), PMLR, pp. 469–477. 3

[BM67] Brigham E. O., Morrow R.: The fast fourier transform. *IEEE spectrum 4*, 12 (1967), 63–70. 2, 3

[CCC*23] Cheng S.-I., Chen Y.-J., Chiu W.-C., Tseng H.-Y., Lee H.-Y.: Adaptively-realistic image generation from stroke and sketch with diffusion model. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2023), pp. 4054–4062. 3, 7, 9

[CJM20] Chi L., Jiang B., Mu Y.: Fast fourier convolution. In *Advances in Neural Information Processing Systems* (2020). 3, 5, 9

[CLGS18] Cui Y. R., Liu Q., Gao C. Y., Su Z.: Fashiongan: display your fashion design using conditional generative adversarial nets. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 109–119. 1, 2, 7, 9

[CMG21] Cao R., Mo H., Gao C.: Line art colorization based on explicit region segmentation. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 1–10. 3

[FCRC05] Fontana M., Carubelli A., Rizzi C., Cugini U.: Clothassembler: a cad module for feature-based garment pattern assembly. *Computer-Aided Design and Applications 2*, 6 (2005), 795–804. 1

[GDNR22] Guo S., Deschaintre V., Noll D., Roullier A.: U-attention to textures: Hierarchical hourglass vision transformer for universal texture synthesis. In *Proceedings of the 19th ACM SIGGRAPH European Conference on Visual Media Production* (2022), pp. 1–10. 3

[GEB15] Gatys L., Ecker A. S., Bethge M.: Texture synthesis using convolutional neural networks. *Advances in neural information processing systems 28* (2015). 3, 10

[GGL22] Gonthier N., Gousseau Y., Ladjal S.: High-resolution neural texture synthesis with long-range constraints. *Journal of Mathematical Imaging and Vision 64*, 5 (2022), 478–492. 3

[GPAM*20] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.: Generative adversarial networks. *Communications of the ACM 63*, 11 (2020), 139–144. 3

[HLBK18] Huang X., Liu M.-Y., Belongie S., Kautz J.: Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 172–189. 3, 7, 9

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems 30* (2017). 7

[HS21] Ho J., Salimans T.: Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications* (2021). 7

[HTB*22] Ham C., Tarrés G. C., Bui T., Hays J., Lin Z., Collomosse J.: Cogs: Controllable generation and search from sketch and style. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI* (2022), pp. 632–650. 3

[HVCB21] Heitz E., Vanhoey K., Chambon T., Belcour L.: A sliced wasserstein loss for neural texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9412–9420. 3, 10

[HZRS16] He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778. 6

[JZYS23] Jain J., Zhou Y., Yu N., Shi H.: Keys to better image inpainting: Structure and texture go hand in hand. In *WACV* (2023). 3, 4, 6

[KB15] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. In *ICLR (Poster)* (2015). 7

[KKL19] Kim B.-K., Kim G., Lee S.-Y.: Style-controlled synthesis of clothing segments for fashion image manipulation. *IEEE Transactions on Multimedia 22*, 2 (2019), 298–310. 2

[KLA*20] Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J., Aila T.: Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR* (2020). 5, 6, 7

[KMR*23] Kirillov A., Mintun E., Ravi N., Mao H., Rolland C., Gustafson L., Xiao T., Whitehead S., Berg A. C., Lo W.-Y., et al.: Segment anything. *arXiv preprint arXiv:2304.02643* (2023). 6

[LGX16] Liu G., Gousseau Y., Xia G.-S.: Texture synthesis through convolutional neural networks and spectrum constraints. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (2016), IEEE, pp. 3234–3239. 3

[LKL*20] Lee J., Kim E., Lee Y., Kim D., Chang J., Choo J.: Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020). 3, 4, 7, 9

[LYH*20] Li Y., Yu X. G., Han X. G., Jiang N. J., Jia K., Lu J. B.: A Deep Learning Based Interactive Sketching System for Fashion Images Design. In *Pacific Graphics Short Papers, Posters, and Work-in-Progress Papers* (2020), Lee S.-h., Zollmann S., Okabe M., Wuensche B., (Eds.), The Eurographics Association. doi:10.2312/pg.20201224. 2

[LZSE21] Liu B., Zhu Y., Song K., Elgammal A.: Self-supervised sketch-to-image synthesis. In *Proceedings of the AAAI conference on artificial intelligence* (2021), vol. 35, pp. 2073–2081. 7, 9

[MLD*20] Mardani M., Liu G., Dundar A., Liu S., Tao A., Catanzaro B.: Neural ffts for universal texture image synthesis. *Advances in Neural Information Processing Systems 33* (2020), 14081–14092. 3, 5, 7

[MSSG*21] Mo H., Simo-Serra E., Gao C., Zou C., Wang R.: General virtual sketching framework for vector line art. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–14. 5

[QZH*19] Qin X., Zhang Z., Huang C., Gao C., Dehghan M., Jagersand M.: Basnet: Boundary-aware salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 7

[RBL*22] Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695. 10

[RGF*20] Reddy P., Guerrero P., Fisher M., Li W., Mitra N. J.: Discovering pattern structure using differentiable compositing. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–15. 12

[SEB*18] Sbai O., Elhoseiny M., Bordes A., LeCun Y., Couprie C.: Design: Design inspiration from generative networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (2018), pp. 0–0. 2

[SLL20] Shen Y., Liang J., Lin M. C.: Gan-based garment generation using sewing pattern images. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2020). 2

[SLM*22] Suvorov R., Logacheva E., Mashikhin A., Remizova A., Ashukha A., Silvestrov A., Kong N., Goka H., Park K., Lempitsky V.: Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (2022), pp. 2149–2159. 3, 6

[SZ15] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015). 5, 6

[WLL*15] Wang K., Lin L., Lu J., Li C., Shi K.: Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence. *IEEE Transactions on Image Processing 24*, 10 (2015), 3019–3033. 7

[XSA*18] Xian W., Sangkloy P., Agrawal V., Raj A., Lu J., Fang C., Yu F., Hays J.: Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8456–8465. 1, 2, 7, 9

[YZL*22] Yan H., Zhang H., Liu L., Zhou D., Xu X., Zhang Z., Yan S.: Toward intelligent design: An ai-based fashion designer using generative adversarial networks aided by sketch and rendering generators. *IEEE Transactions on Multimedia* (2022). 2

[ZCS*21] Zhao S., Cui J., Sheng Y., Dong Y., Liang X., Chang E. I., Xu Y.: Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)* (2021). 5, 7, 9

[ZIE*18] Zhang R., Isola P., Efros A. A., Shechtman E., Wang O.: The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR* (2018). 7

[ZLL*22] Zheng H., Lin Z., Lu J., Cohen S., Shechtman E., Barnes C., Zhang J., Xu N., Sohrab A., Luo J.: Cm-gan: Image inpainting with cascaded modulation gan and object-aware training. In *Proceedings of the European conference on computer vision (ECCV)* (2022), pp. 277–296. 3

[ZZB*18] Zhou Y., Zhu Z., Bai X., Lischinski D., Cohen-Or D., Huang H.: Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics (Proc. SIGGRAPH) 37*, 4 (2018). 3

[ZZP*17a] Zhou B., Zhao H., Puig X., Fidler S., Barriuso A., Torralba A.: Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017). 6

[ZZP*17b] Zhu J.-Y., Zhang R., Pathak D., Darrell T., Efros A. A., Wang O., Shechtman E.: Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems* (2017). 2

[ZZP*18] Zhou B., Zhao H., Puig X., Xiao T., Fidler S., Barriuso A., Torralba A.: Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision* (2018). 6