CoDE-GAN: Content Decoupled and Enhanced GAN for Sketch-guided Flexible Fashion Editing

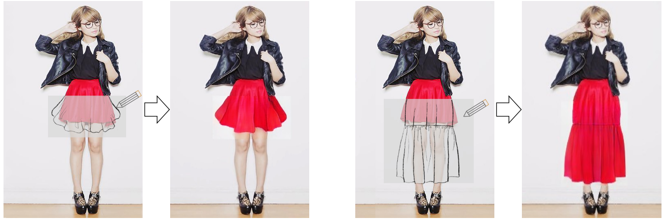
Zhengwentai Sun, Yanghong Zhou and P. Y. Mok, *Member, IEEE*

***Abstract*—Rapid advancements in generative models, including generative adversarial networks (GANs) and diffusion models, have made possible of automated *image editing* through the use of text descriptions, semantic segmentation, and/or reference style images. Nevertheless, in terms of fashion image editing, it often requires more flexible, and typically iterative, modifications to the image content that existing methods struggle to achieve. This paper proposes a new model called Content Decoupled and Enhanced GAN (CoDE-GAN), which is formulated and trained for the task of image reconstruction, more specifically, image inpainting with sketch-guidance. Through this proxy task, the trained model can be used for flexible image editing, generating new images with consistent colours and required textures based on sketch inputs. In this new model, a content decoupling block is introduced including specially designed dual encoders, which pre-process inputs and transform into separated structure and texture representations. Moreover, a content enhancing module is designed and applied to the decoder, improving the colour consistency and refining the texture of the generated images. The proposed CoDE-GAN can achieve coarse-to-fine results in one single stage. Extensive experiment on three datasets, covering human, garment-only and scene images, show that CoDE-GAN outperforms other state-of-the-art methods in terms of both generated image quality and editing flexibility. The code will be released once the paper is published.**

***Index Terms*—Fashion image editing, content decoupling, content enhancement, GAN-based.**

I. Introduction

IMAGE editing has drawn great attention in the digital era and is used in many tasks such as removal of unwanted objects and adjusting style. Typically, image editing is completed by the use of professional software or tools (e.g., Adobe Photoshop, Fotor Photos). The use of such software or tools, however, requires professional knowledge, and the process is also tedious, time-consuming and skill dependent. Benefited from the success in generative adversarial networks (GAN), image editing can now be automated using inputs such as text descriptions [1]–[3], semantic segmentation [4], [5], and reference style images [6]. GAN-based methods are recently developed for interesting applications of fashion image editing, for example, fashion images are edited into different poses by pose-guided image synthesis [7], [8] or different clothing are 'tried-on' on users bodies based on their reference photos [9], with astonishing performance in terms of geometric shape or texture transformation.



(a) Edit to a Circular Skirt (b) Edit to a Tiered Skirt

Fig. 1. The user may have various and changeable demands on a fashion garment. For example, they may be satisfied with the skirt color but want to have different styles. Our proposed CoDE-GAN enables the user to draw simplified sketches to flexibly edit the clothing shape with consistent textures.

Nevertheless, most image editing tasks in the fashion domain require more flexible and controllable modifications to the target fashion images. Fig. 1 illustrates a scenario that is common in fashion design or fashion presentation, in which a user wants to make some minor modifications to the partial region of the skirt, like changing its type to be a circle skirt (Fig. 1 (a)) or changing its length to be a tiered skirt (Fig. 1 (b)). Those above-mentioned existing GAN-based methods are not suitable. In order to facilitate users to freely edit their ideas on the image, sketch-guided methods were proposed recently [10]–[14] and received more and more attention. These approaches are based on sketch-guided image inpainting techniques that reconstruct the masked image I_M with sketch S and mask M as reference. When editing an image, the user could draw their preferred sketch to modify the shape. However, it is still challenging to achieve sketch-controlled editing via the sketch-guided inpainting approach. Although the sketch provides guidance for the structural information, it mainly captures the boundary and lacks detail in the inner region. As a result, the synthesized textures become less plausible and exhibit inconsistencies with the unmasked region. This problem is particularly pronounced when dealing with large masked regions or complex input sketches, making it even more challenging for the model to generate images with consistent structure and texture.

To address this issue, E2I [11] adopts a coarse-to-fine architecture, applying a contextual attention mechanism to improve the synthesized textures in the fine stage. Gated Conv [10] learns a soft mask to weight different regions, improve the model's ability in inferring the missing content by referring to the unmasked region. However, their methods never consider

The work described in this paper was supported in part by the Innovation and Technology Commission of Hong Kong under grant ITP/028/21TP and by the Research Institute of Intelligent Wearable Systems under grant P0049355.

All authors are with School of Fashion and Textiles of The Hong Kong Polytechnic University; Y. H. Zhou is also with the Research Centre of Textiles for Future Fashion of The Hong Kong Polytechnic University; and P. Y. Mok is also with The Research Institute of Intelligent Wearable Systems of The Hong Kong Polytechnic University.

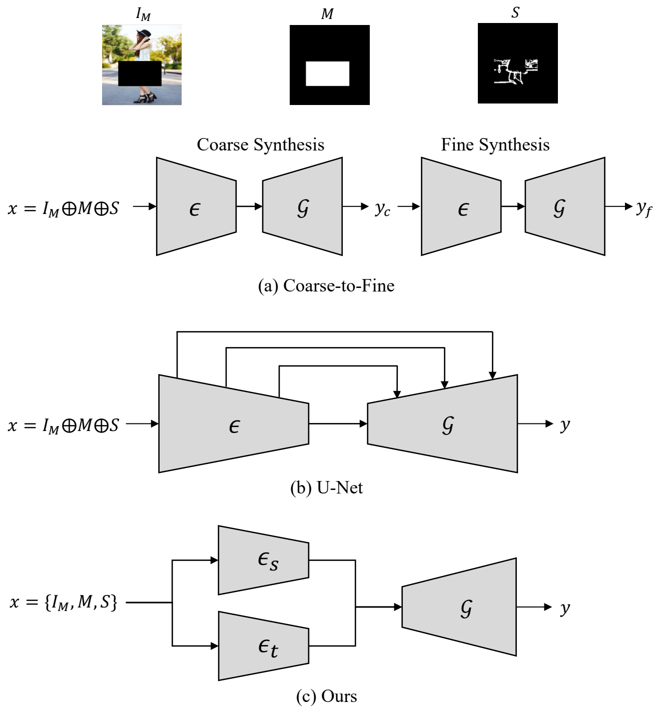
Corresponding author: P. Y. Mok (tracy.mok@polyu.edu.hk)

to effectively utilize sketch to guide the synthesis. Instead, they explicitly concatenate the sketch to learn a synthesis. Fig. 2 (a) illustrates this. Instead of simply concatenation, DeFLOCNet [14] argues that the sparse sketch may vanish through the network layers. Therefore, they propose to insert sketch into every skip connections between encoder and decoder. Fig. 2 (b) shows their ideas. Their approach could emphasize the importance of sketch but less effective in inferring sketch-contoured textures. Recently, ControlNet [15] introduced an image generation model based on diffusion models [16]–[19], which can use sketches to control the spatial structure of synthesized images. By integrating ControlNet with the inpainting model of Stable Diffusion [20], it becomes possible to perform sketch-guided image editing tasks. However, ControlNet heavily relies on textual descriptions for synthesizing image textures. In the absence of a text prompt, as in this specific task, ControlNet struggles to produce consistent textures.

In this paper, we propose a novel model, **C**ontent **D**ecoupled & **E**nhanced **GAN** (CoDE-GAN) to address these issues. Different from previous work, our proposed model decouples the image content into structure and texture representations through a Content Decoupling Module (CDM). Specifically, a condition decoupling block (CDB) is first used to obtain the structure and texture conditions from the input set x. Then, different encoders are utilized to learn the specific representations for the structure and texture conditions. Fig. 2 (c) shows a simplified structure of our decoupling idea. There are two advantages to doing this: (1) The encoder ϵ_t only handles texture conditions and enables the model to focus more on texture synthesis and learn better texture representations. (2) The structure encoder ϵ_s benefits the latter synthesis process by obtaining structure representation that is distinct from texture representation. These enhanced representations can then help to generate better realistic images having a consistent structure with the input sketch and reasonable texture with the unmasked region. In addition, to further improve the consistency of textures within the content region, we add a Content Enhancement Module (CEM) to the generation decoder. Fig. 3 shows the detailed architecture. The CEM extracts intermediate features from the decoder and transforms them into a single feature map, and then adds a constraint loss to constrain the similarity between the feature map and the grey image. If the synthesized textures are consistent with unedited region, the content response should be similar in feature level.

In conclusion, our contributions are as follows:

1) We propose a novel network for flexible editing of a fashion image generating content with consistent textures and sketches.
2) The content decoupling module we designed could faithfully obtain sketch and texture representations which lead to robust performance on flexible editing.
3) The content enhancement module we designed could improve the consistency of synthesized content.
4) Extensive experiments are conducted to demonstrate the performance of our proposed CoDE-GAN on four datasets, including fashion human ATR dataset [21], in-shop Garment dataset [22], CafiDataset and LSUN outside church dataset [23]. On all these datsets, our methods achieves good and robust performance in FID, SSIM and PSNR metrics.



(a) Coarse-to-Fine

(b) U-Net

(c) Ours

Fig. 2. Simplified network structure of the explicitly concatenation (a), sketch injection (b), and ours decoupling (c).

II. Related Work

A. Fashion Image Editing

Fashion image editing tasks could be classified, according to the target objects being edited, as *human-centric editing* and *garment-centric editing*.

a) Human-centric Editing: This task primarily addresses two major challenges: virtual try-on and pose transfer. Virtual try-on requires transferring a source cloth to the target human body. While the latter, pose transfer, is about synthesizing consistent human images across a range of poses.

VITON [24] is the first work that considered image-based virtual try-on. They tackled the misalignment issue between source clothing and the target human pose by using TPS [25] to adapt the cloth according to key point correspondences. Moreover, they introduced a refinement network that predicts a composition mask, locating the exact region for try-on. To preserve the characteristics of the source cloth, Wang *et al.* [26] introduced CP-VTON, which employs a Geometric Matching Module (GMM) to determine the TPS transformation based on human parsing and pose data. LA-VITON [27] leveraged the StyleGAN [28] structure, directing the generator to emphasize local clothing deformation, thereby enhancing the synthesized textures. Despite the significant results achieved by the previous work, they are less effective in addressing reasonable try-on results in self-occlusion regions (e.g., crossed arms). To address this issue, Yang *et al.* addressed this in ACGPN [29] by predicting post-try-on human parsing results, instead of the conventional composition mask. To allow seamless try-on results, they further introduce an inpainting module that fills in the area where the warped cloth will go.

Another approach to address complex pose and self-occlusion is considering pose transfer. Ren *et al.* [30] introduced a global flow framework that utilizes pose data to predict flow between source and target images. Subsequently, feature patches from the source image can be located in the target image through their proposed attention module. Han *et al.* [31], on the other hand, predicts this flow based on a human parsing map. Cui *et al.* [32] combined elements of both virtual try-on and pose transfer, resulting in a comprehensive 'dressing in order' framework. Their approach begins by generating a pose-transferred image and subsequently overlays clothes in a predetermined sequence. This method can simultaneously achieve virtual try-on and pose transfer.

To further enhance the resolution of synthesized images, Choi *et al.* [33] propose the VITON-HD dataset and method that could first synthesize high-resolution virtual try-on images. In Lee *et al.*'s work, their HR-VITON unifies the previous separated warp module and human parsing map that after try-on synthesis into a single condition generator. Their method could effectively consider the relationship between parsing map and warped cloth and therefore preserve the cloth detail. Recently, with the achievement of diffusion models [16]–[19], [34], Zhu *et al.*'s TryOnDiffusion [35] could achieve detail-preserved cloth warping and high-resolution try-on results by their diffusion model. Different from TryOnDiffusion, LaDI-VTON [36] takes the warped cloth as an auxiliary input and inverts the unwarped cloth into semantic space. Therefore, the try-on results could be controlled not only by the cloth image but also by a text description.

While the aforementioned approaches have shown impressive outcomes, they come with inherent limitations. They demand multiple inputs like human parsing, pose estimation, text descriptions, etc. Such extensive input requirements not only increase the computational overhead but also limit the flexibility of image editing. For shape editing, like in Fig. 1, users first find a desired in-shop cloth and then utilizes the above-mentioned models to try it on, making the editing process less efficient.

b) Garment-centric Editing: Dai *et al.* [37] argued that it is important to edit design drafts. Their fashion editing workflow formulates the fashion editing task as a bidirectional image translation task. By translating an in-shop fashion garment to design draft, it benefits the designer in making modifications. Their pipeline is able to translate the edited drafts back into new in-shop garments. TailorGAN [22] achieves fashion attribute editing by specifying a reference image. To address the lack of paired data between input garments and edited images, TailorGAN proposes a self-supervision training pipeline. By reconstructing a masked attribute region with the guidance of a reference image, TailorGAN could process fashion editing tasks. Nevertheless, this method can only be applied to limited local areas like editing collars and sleeves, which leads to poor generalization to other attributes. Even though the existing work are capable of editing fashion garments to some extent, it is demanding to provide a user-friendly interaction in editing in-shop clothing.

B. Sketch-Guided Image Inpainting

Image inpainting recovers a masked region with consistent context to a valid region. It assumes that the masked location is given. Due to the loss of information, it is challenging to recover consistent structure and texture. Therefore, Nazeri *et al.* [38] proposed an edge-connect way for first reconstructing the sketch map in the damaged region. With prior information, the recovered edge map contributed to the completion of the task. Their edge-connecting pipeline enables user-guided editing and could achieve better semantic consistency with the help of inpainted edges. However, their methods explicitly take a masked edge map as input. This is against to real inpainting scenario that the edge map could only been obtained from a damaged image. Directly applying edge detection would turn the masked boundary as edges. The boundary edges would be inconsistent with the clean masked edge map and may lead to the degradation to arbitrary image editing. Therefore, Xu *et al.* [11] proposed a three-stage network E2I that utilizes sketches to assist in the inpainting process. In the initial stage of the E2I, it inpaints the sketches within the missing areas. These approximated sketches are then fed into the second and third networks, following a coarse-to-fine approach. Unlike Edge-connect [38], which assumes that the damaged edge has been obtained, E2I directly employs its coarse-to-fine inpainting network with an empty edge as the input. Subsequent to this, edge detection is performed on the resulting images, and the corresponding areas are masked to produce the damaged edge map. Edgeconnect then integrates this damaged edge map back into their entire pipeline again. Although E2I could utilize the edge map to guide the inpainting process, their pipeline requires a three-stage network and costs much memory and time to inference.

Other edge-guided inpainting work treat the edge as existing additional auxiliary information. In Gated Conv [10], they utilize a coarse-to-fine approach similar to the above-mentioned E2I [11], but with a significant modification: they replace the conventional convolution process with their proposed gated convolution to obtain soft gating weights. This soft gating is realized by introducing an extra convolution process to predict. Their gated convolution is robust in processing free-form masks (random strokes). In addition to the gated convolution, they also integrate an edge map directly into their model so as to guide the inpainting process. Following Gated Conv, Jo [12] introduced SC-FEGAN for addressing face editing tasks. They consider the editing process as image-to-image translation [39], choosing the U-Net structure instead. Their model can modify not just the image shape but also the color by incorporating a user-guided color map as an input. Yang *et al.* [13] consider the discrepancy between detected edges and human-drawn edges by utilizing an edge refinement network before the edge-guided inpainting network. To avoid the edge information diminishing in the feature space, DeFLOCNet [14] utilizes structure generation blocks to inject the edge into each skip connection in a U-Net.

Although the above-mentioned approaches could edit an image through learning a sketch-guided inpainting proxy task,

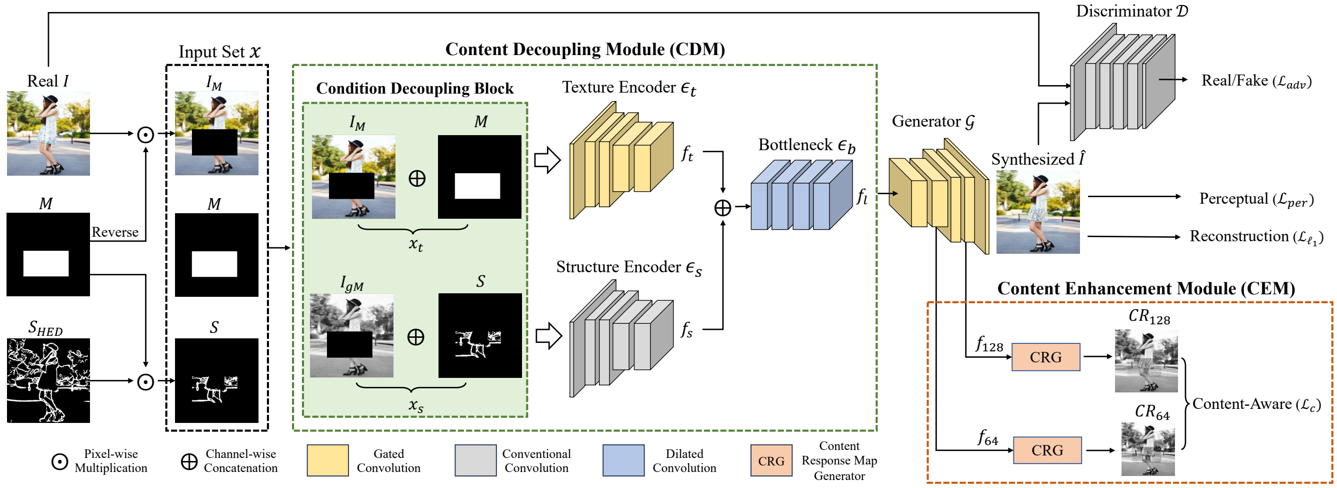

Discriminator $\mathcal{D}$
Real/Fake ($\mathcal{L}_{adv}$)
Input Set x
Content Decoupling Module (CDM)
Real I
I_M
Condition Decoupling Block
Texture Encoder ϵ_t
Generator $\mathcal{G}$
I_M
M
f_t
Bottleneck ϵ_b
Synthesized $\hat{I}$
f_l
M
M
Reverse
x_t
Perceptual ($\mathcal{L}_{per}$)
Reconstruction ($\mathcal{L}_{\ell_1}$)
Structure Encoder ϵ_s
Content Enhancement Module (CEM)
I_{gM}
S
S_{HED}
S
f_s
CR_{128}
f_{128}
CRG
x_s
Content-Aware ($\mathcal{L}_c$)
CR_{64}
f_{64}
CRG
Pixel-wise Multiplication
Channel-wise Concatenation
Gated Convolution
Conventional Convolution
Dilated Convolution
CRG
Content Response Map Generator

Fig. 3. An overview of our proposed CoDE-GAN. It incorporates Content Decoupling Module to obtain latent representation f_l of the input set x. In the latter generation process, a Content Enhancement Module is applied to further improve the consistency between the synthesized textures and the unedited textures.

they often overlook the gap between editing and inpainting. In editing, user-provided masks roughly determine the editing area. But when editing larger regions, the mask may cover a continuous space, similar to a box-shaped mask. Few work have ever studied the impacts of different masks, e.g. between free-form and box masks. Although Edgeconnect [38] and E2I [11] have considered the effect of mask ratios, they have not explored the influence of different mask shapes. Regarding the input edge, only the work by Yang *et al.* [13] addresses the shape differences between detected edges and those drawn by humans. However, the effectiveness of binarization remains unexplored in any of these studies.

Benefited from the outstanding capability of Stable Diffusion [20] in synthesizing images from textual descriptions, ControlNet [15] replicates and fine-tunes the encoder from Stable Diffusion to accept sketches as input. Consequently, ControlNet can generate images that reflect the structure of the input sketches and the textures of the input texts. Nevertheless, it struggles to synthesize textures that are consistent with the unedited regions unless detailed textual descriptions of the textures are provided.

III. Method

Our proposed CoDE-GAN includes a **C**ontent **D**ecoupling **M**odule (CDM) and a **C**ontent **E**nhancement **M**odule (CEM). In this section, we will first provide the problem formulation for the sketch-guided image editing. Then we will give the details of specific modules. We will also introduce the optimizing objectives.

A. Problem Formulation

The aim of sketch-guided image editing is to synthesize an image with user-intended sketches. Let $I \in \mathbb{R}^{3\times w\times h}$ be the ground truth RGB image where w is the image width and h is the image height, $M \in \mathbb{R}^{1\times w\times h}$ be the binary mask where 1 indicates editing or masked area and 0 indicates the unmasked region, and S be the input sketch, the sketch-guided image editing model will generate a new image which is filled in the consistent texture in the masked region M and has the consistent sketch with S. During the training stage, sketch S is extracted by edge detection network HED [40] $H(\cdot)$ and multiplied with the mask M, which can be defined as:

$$S = H(I) \odot M, \tag{1}$$

where $\odot$ is the element-wise multiplication. Since HED can only output a greyscale sketch map, S is binarized by setting the threshold to 0.6 to simulate users' drawn sketches. During the inference stage, S is drawn by the users in the editing area. In general, the inputs of the sketch-guided image synthesis are the set $x = \{I_M, M, S\}$, where I_M is the masked RGB image obtained by:

$$I_M = I \odot (1 - M) \tag{2}$$

To make the model learn specific texture and structure representation for better image synthesis, the CDM is designed to learn the decoupled texture representation and structure representation and fuse them to obtain better latent representation for image generation. Let the latent representation be f_l, it can be represented by:

$$f_l = \mathrm{CDM}(x) \tag{3}$$

The latent representation is then fed into a generator $\mathcal{G}$ to generate a synthesized image, which is defined by:

$$\hat{I} = \mathcal{G}\,(f_l) \tag{4}$$

Lastly, four loss functions are used to train the network to make the synthesized image $\hat{I}$ similar to the original image I as much as possible. The detail of the loss functions is illustrated in Section III-E.

B. Content Decoupling Module

The content decoupling module consists of a Condition Decoupling Block (CDB), a structure encoder, a texture encoder, and a bottleneck.

a) Condition Decoupling Block: This block decouples the input x into two types of conditions: the texture condition x_t and the structure condition x_s. Given the image I, the mask M and the sketch S, the x_t and x_s can be computed by:

$$x_t = I_M \oplus M \tag{5a}$$

$$x_s = I_{gM} \oplus S, \tag{5b}$$

where $\oplus$ is channel-wise concatenation, and $I_{gM} \in \mathbb{R}^{1\times w\times h}$ is the grey image of I_M. It can be seen from the formulas that the input $x_t \in \mathbb{R}^{4\times w\times h}$ aligns the setting of image inpainting and the input $x_s \in \mathbb{R}^{2\times w\times h}$ is conditioned to the sketch. Here, x_s incorporates sketch with the grey image instead of RGB image, because grey image is more effective to represent structural information than RGB image and reduces the representation space from $\mathbb{R}^3$ to $\mathbb{R}^1$. Moreover, traditional image processing algorithms, such as Canny edge detection, typically work with grey images to obtain edge details.

b) Texture Encoder: The texture encoder ϵ_t feeds in the condition x_t and learn the texture representation by:

$$f_t = \varepsilon_t\left(x_t\right), \tag{6}$$

where ϵ_t is the texture encoder. As the texture encoder mainly aims to reconstruct the texture of the masked region, which is the same as the image inpainting task, we adopt the encoder structure of Gated Conv [10]. Gated Conv designs a gated convolution that adapts a dynamic feature selection mechanism to make the convolution dependent on the soft mask that is automatically learned from data, and improves the texture consistency and inpainting quality of the masked region. Specifically, for the input feature f_{in}, a gated convolution $Conv_g$ applies an additional convolution to obtain a soft weight map and then multiples it with a learned feature of f_{in}. It is formulated as:

$$\operatorname{Conv}_g\left(f_{\text{in}}\right) = \operatorname{Conv}\left(f_{\text{in}}\right) \odot \sigma\left(\operatorname{Conv}_d\left(f_{\text{in}}\right)\right), \tag{7}$$

where Conv is the conventional convolution, $\operatorname{Conv_d}$ is the convolution that outputs single-channel feature map, and σ is the sigmoid function that scales learned gating to range $(0, 1)$.

c) Sturcture Encoder: The structure encoder ϵ_s takes the input x_s and learns the structure representation f_s by:

$$f_s = \varepsilon_s\left(x_s\right) \tag{8}$$

The structure of ϵ_s is same with ϵ_t, but the gated convolution is replaced with conventional convolution. There are two reasons why we use conventional convolution here: 1) we wish this encoder mainly focused on capturing the basic structure of the whole image, and thus the texture information learning is not that important and will be achieved by the texture encoder. 2) Gated convolution adapts an extra convolution to learn the soft weighting map, leading to an increase in computation cost.

d) Bottleneck: Lastly, we fuse the texture representation and structure representation by a bottleneck structure to reduce the representation space. The bottleneck structure consists of four dilated gated convolution blocks. We firstly concatenate f_t and f_s, and feed it to a bottleneck ϵ_b to object the fused latent representation f_l. It is formulated as:

$$f_l = \epsilon_b(f_t \oplus f_s) \tag{9}$$

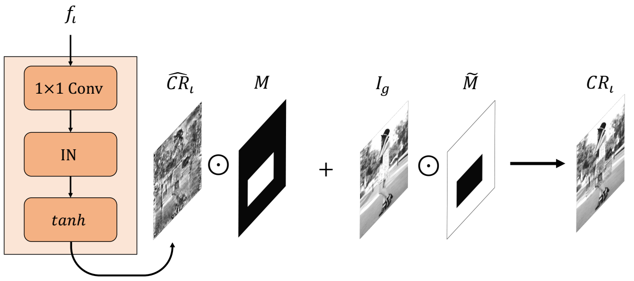


Fig. 4. Content response map generator (CRG) transforms features into content response map. The response map is masked and fused with a grey image.

C. Adversarial Generation

To allow the synthesized results more realistic and reasonable, the adversarial generation process is incorporated.

a) Generator: Given the fused latent representation f_l, the generator $\mathcal{G}$ could synthesizes a fake image $\hat{I}$:

$$\hat{I} = \mathcal{G}(f_l) \odot M + I_M \tag{10}$$

The $\mathcal{G}$ consists of five gated convolution blocks with twice upsampling which is symmetric to the structure of encoder ϵ_t.

b) Discriminator: Following with Pix2Pix [39], we implemented a patch discriminator $\mathcal{D}$ which output real/fake discrimination on image patches instead of the whole image. Its discrimination could focus on local details and enhance the fidelity of the generated image. The structure of $\mathcal{D}$ is like an encoder that only consists of six convolution blocks. Besides, to stabilize the adversarial training process, we adopted spectral normalization on the discriminator as well [41].

D. Content Enhancement Module

To further improve the consistency of synthesized content, a Content Enhancement Module (CEM) is applied to the generator $\mathcal{G}$. As shown in Fig. 3, CEM extracts the features from the second and fourth blocks. The features have different resolutions and are denoted as f_{64} and f_{128} of which the subscript indicates the resolution of the feature map. Then, the two features are respectively fed into a **C**ontent **R**esponse Map **G**enerator (CRG) to generate the content response maps CR_{64} and CR_{128}. As Fig. 4 illustrates, we obtain the content response map CR_i by:

$$\begin{aligned} CR_i &= CEM(f_i) \\ &= \tanh\left[\text{IN}(\text{Conv}_\text{d}(f_i))\right] \odot M + I_g \odot (1 - M), \end{aligned} \tag{11}$$

where $i = 64$ or $i = 128$, $\operatorname{Conv_d}$ reduces the feature dimensionality of f_i to single, IN denotes an instance normalization layer, and $\tanh$ is a Tanh activation function. Then, we calculate the cosine similarity between CR_i and the grey image I_g and regard it as an objective function, which is computed by:

$$\mathcal{L}_c = \left(1 - \frac{CR_{64} \cdot I_g}{\|CR_{64}\| \, \|I_g\|}\right) + \left(1 - \frac{CR_{128} \cdot I_g}{\|CR_{128}\| \, \|I_g\|}\right) \tag{12}$$

We hope that by minimizing the similarity distance between the CR_i and the grey image I_g, the features of generator can also be optimized through gradient backpropagation.

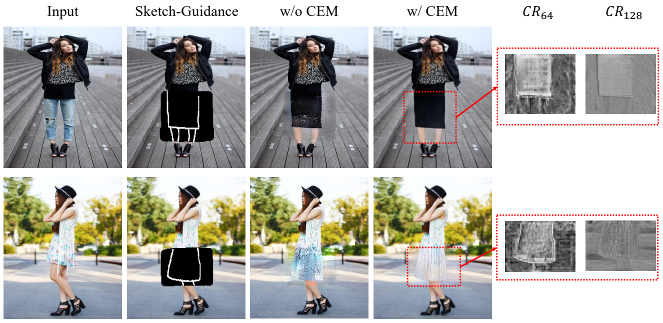


Fig. 5. Visualization of the synthesized content response map CR at resolution of 64×64 and 128×128.

We visualize content response maps at resolutions 64×64 (CR_{64}) and 128×128 (CR_{128}) in Fig. 5. It could be observed that the CEM could learn the structure and texture of the image and the content response map with a higher resolution clearly exhibits more uniform content and sharper boundaries. Since the input sketch is sparse and gradually diminishes in the CNN feature space, it is important to inject the sparse sketch information in the CNN space, especially in the generator. In DeFlocNet [14], the control inputs are injected in all blocks of encoders and generators to preserve the guidance information. However, this method will add additional computation costs and can not provide other content information except the input controls, like the structure and texture information around the sketch. In our case, we optimize the features of the generator to be like the original grey image which has rich structure and texture information, the generator will learn to recover the structure and texture of the masked region as shown in Fig. 5. Therefore, our proposed CEM is able to enhance and refine the content information, leading to more detailed and high-quality generation results.

E. Optimization Objectives

For training the CoDE-GAN, except for the above-mentioned content-aware loss, we use reconstruction loss, perceptual loss, and generative adversarial loss. In the following, we will introduce these loss functions.

a) Reconstruction Loss: To ensure the generated image $\hat{I}$ is close to the RGB image I within the unmasked region, L1 loss is used between them on the unmasked region. It is defined by:

$$L_{\ell 1} = |I - \hat{I}|_1 \odot M \tag{13}$$

b) Perceptual Loss: Following style transfer, we introduce perceptual loss [42] to keep the perceptual information as well. It is obtained by:

$$\mathcal{L}_{per} = \sum_i w_i \cdot L1 \left(F_i(I) - F_i(\hat{I}) \right), \tag{14}$$

where F_i stands for ith activation layer of VGG-19 network, and w_i is the corresponding weight. Specifically, the selected layers are *relu1_1, relu2_1, relu3_1, relu4_1* and *relu5_1*. In our experiments, we set the all corresponding weight w_i as 1.0.

c) Generative Adversarial Loss: The synthesis process is conditioned to inputs $x = \{I_M, M, S\}$. To allow the discriminator $\mathcal{D}$ to consider the conditions, despite the real/fake image I and $\hat{I}$, $\mathcal{D}$ will take x as well. We adopted hinge loss for optimizing spectral normalized discriminator $\mathcal{D}$:

$$\begin{aligned} \mathcal{L}^D_{adv} = & \mathbb{E}_{I,x}[\min(0, -1 + \mathcal{D}(I, x))] + \\ & \mathbb{E}_{\hat{I}_x, x}[\min(0, -1 - \mathcal{D}(\hat{I}, x))] \end{aligned} \tag{15}$$

And the adversarial loss for the total network CoDE-GAN:

$$\mathcal{L}^G_{adv} = -\mathbb{E}_{\hat{I},x}[\mathcal{D}(\hat{I}, x)] \tag{16}$$

The overall objectives are:

$$\mathcal{L} = \lambda_{\text{per}} \mathcal{L}_{\text{per}} + \lambda_{\ell 1} L_{\ell 1} + \lambda_c \mathcal{L}_c + \mathcal{L}^G_{adv}, \tag{17}$$

$\lambda_{per}, \lambda_{\ell 1}, \lambda_c, \lambda_{adv}$ denotes the coefficients for perceptual loss, reconstruction, content-aware loss and adversarial loss respectively.

IV. Experiments

We conducted extensive quantitative and qualitative experiments on several datasets, such as fashion human, garment, and outdoor church dataset, to demonstrate the effectiveness of our proposed CoDE-GAN. In this section, we first introduce the dataset and experiments settings. Then, we compared with the state-of-the-art methods and conducted ablation studies of the proposed CDM and CEM modules. Lastly, we further discuss the influence of edge pre-processing and mask types. It shows that our methods are robust.

A. Dataset and Experiments Settings

a) Datasets: We investigate the effectiveness of our model on two garment synthetic datasets (Garment dataset [22] and CafiGarment dataset), a fashion human dataset (ATR dataset [21]), and an outdoor building dataset (LSUN outdoor church [23]): (1) Garment dataset consists of 9.6k images about upper clothing. (2) CafiGarments collected 17k images with 79 clothing categories across upper, bottom, and full body clothing. (3) ATR dataset comprises of 17.7k humane images with various poses and complex background. (4) LSUN outdoor church is a subset of LSUN dataset which consists of 126k images. During our experiments, these above-mentioned datasets are divided into train and test set with the ratio of 9:1. Except the LSUN outdoor church dataset that we adopt its official validation set for testing which contains 300 images.

b) Evaluation Metrics: Measuring the quality of edited image content in a quantitative way can be challenging. Collecting pairs of edited images for comparison can be time-consuming and expensive. Therefore, to evaluate the editing results in a more practical and cost-effective way, we evaluate sketch-guided image inpainting results on test dataset. We employ the Fréchet Inception Distance (FID), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) as our quantitative metrics. FID evaluates the distance between the distribution of the generated images and the ground truth images. By calculating the Fréchet distance in Inception Net's

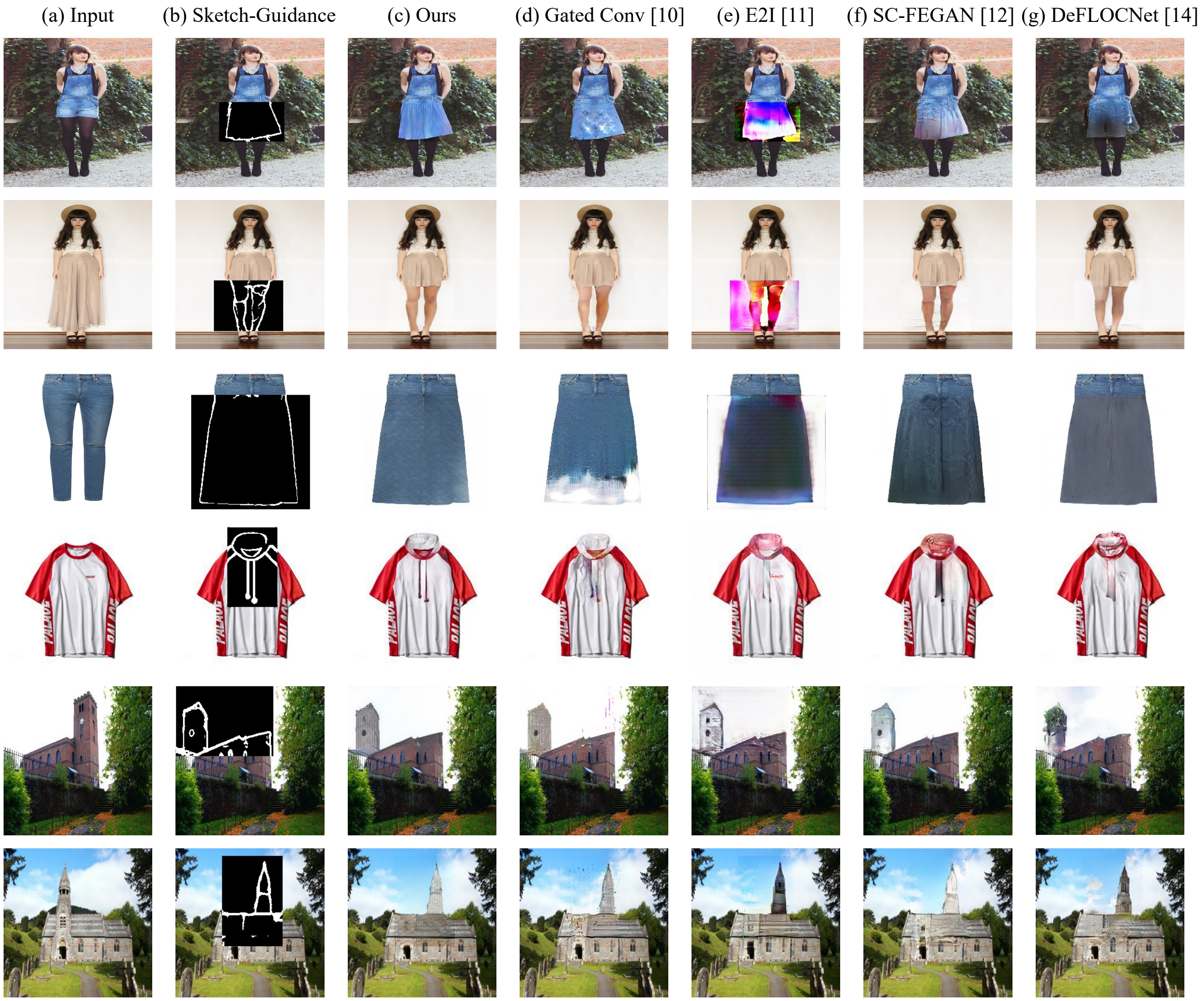


Fig. 6. Comparisons with state-of-the-art methods: Gated Conv [10], E2I [11], SC-FEGAN [12], and DeFLOCNet [14]. The first two rows show the qualitative results in ATR Dataset. The third and fourth rows plot the results of CafiDataset and Garment Dataset. The last two rows present object shifting of the tower part in LSUN outdoor church dataset.

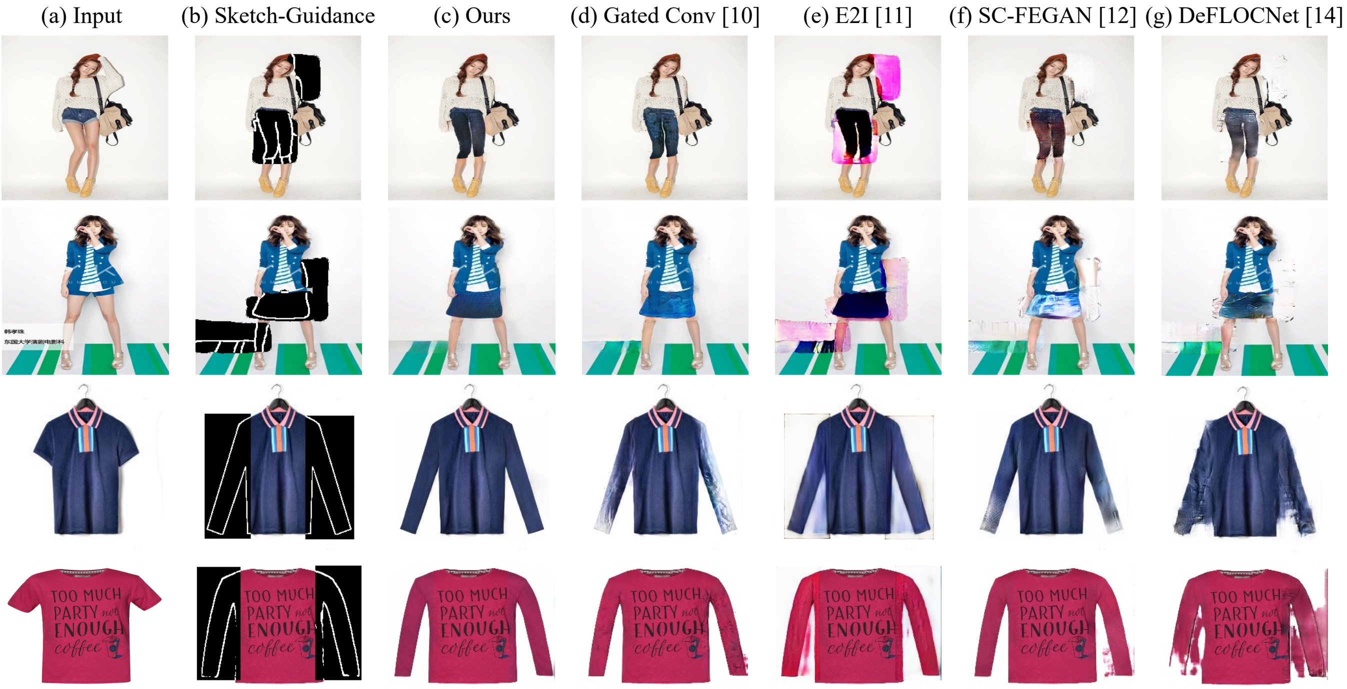
feature space, FID effectively captures the perceptual similarity between the two distributions, with a lower FID indicating higher realism and perceptual similarity. PSNR employs pixel-wise differences to compute a ratio between the maximum possible power of a signal and the power of corrupting noise, serving as a measure of the visibility of errors. SSIM, measures the structural similarity between the generated image and the ground truth image. SSIM is similar to PSNR but is believed better aligned with human perception. Higher values of SSIM and PSNR indicate better image quality. Given our primary aim of editing images rather than reconstructing them, we have selected FID as our principal metric.

c) Implementation Details: Due to computation limitations, we resize each image with resolution of 256×256. We then utilize HED [40] edge detector to obtain sketches. For simulating manually drawn sketches, we binarize the detected edge maps with a threshold of 0.6. In terms of mask generation, we randomly generate single rectangular box mask with a ratio of 30% in all experiments. For loss weights, we set $\lambda_{\ell 1} = 100, \lambda_{\text{per}} = 20, \lambda_c = 5, \lambda_{adv} = 1$ in all experiments. We run all experiments with batch size 12 in a single RTX 3090 GPU. We train 500k iterations in each dataset. We adopt Adam optimizer with learning rate of 1e-4 for the whole synthesis model and 4e-4 for the discriminator.

B. Comparison with State-of-the-Art Methods

We compare our CoDE-GAN with two approaches that utilizing coarse-to-fine structure (E2I, and Gated Conv) and two pixel translation structure (DeFLOCNet, SC-FEGAN).

- Gated Conv [10]: For the implementation of gated conv, we adopted the implementation of PyTorch version and keep the hyper parameters consistent with the original implementation.



(a) Input (b) Sketch-Guidance (c) Ours (d) Gated Conv [10] (e) E2I [11] (f) SC-FEGAN [12] (g) DeFLOCNet [14]

Fig. 7. Flexible editing results on challenging image content and area. The first two rows shows the editing of pose, clothing styles on fashion human. The last two rows edits the two sleeves simultaneously.

- E2I [11]: The E2I proposed three-stage method which consists of edge inpainting network, and a coarse-to-fine network. In the sketch-guided inpainting, we assume the sketch is a given input. Therefore, we removed its edge inpainting part and trained the rest with consistent hyper parameters and losses.
- DeFLOCNet [14]: DeFLOCNet utilized an encoder-decoder structure but choose to inject sketch into each skip connections. We trained it with its original hyper parameters.
- SC-FEGAN [12]: The original SC-FEGAN utilizes color sketches to edit an images additionally. For fair comparison, we only keep the edge sketches as the input.

1) Qualitative Comparison: Fig. 6 illustrates the qualitative comparison results with the above methods on four datasets. As shown in Fig. 6 (e), we can see that E2I has relatively poor ability to generate the reasonable image on the garment datasets and totally collapses on the ATR dataset which consists of the person images. This is because that E2I directly employs traditional convolution operation to the encoder-decoder architecture. On the one hand, the encoder-decoder network processes features through every layer, lacking direct connections between different layer. This lack of inter-layer connectivity restricts its ability to leverage low-level information effectively, limiting its capacity to generate detailed local textures. On the other hand, the traditional convolution operations cannot distinguish between valid and invalid input pixels, resulting in an undesirable blending of conditional information and generating synthesized results with blurred boundaries. Especially for the clothing images which have various styles, material textures and colors, it is much more difficult to infer much from the valid area to fill in the accurate content for masked region. Therefore, E2I performs relatively poor on the image generation on the Garment datasets. Different from E2I, DeFLOCNet adopts the U-Net architecture and add skip connections between the mirrored layers in the encoder and decoder stacks, allowing low-level information to flow pass across the network and can generate more realistic images than the simple encoder-decoder network. So DeFLOCNet generates better results than E2I. However, DeFLOCNet still uses the traditional convolution operations in the network and fails to generate realistic images. SC-FEGAN and Gated Conv both use gated convolution operations in the network. They can generate realistic textures for the things without much variations well, like the skin textures (see 2nd row of Fig. 6) and building textures (see 5th and 6th row of Fig. 6). However, the generation results on the garments still have flaws. For example, SC-FEGAN cannot generate consistent textures with the current clothing (see Fig. 6 (f)) while Gated Conv fail the accurate texture synthesis at some pixels (see Fig. 6 (d)). However, our model employs a Content Decoupling Module to learn better texture and content representation and uses a Content Enhancement Module to supervise the consistency of synthesized textures, our proposed CoDE-GAN generates realistic images with consistent structures and fine-grained texture details on the garment datasets and building dataset.

Fig. 7 exhibits the flexibility of fashion image editing, where all methods are trained using randomly generated single box masks that cover 30% of the image area. By flexibility, we refer to the capability to arbitrarily edit any region regardless of its shape, are, or continuity. More than one mask is used in Fig. 7's samples, demonstrating our method's capacity

TABLE I
Quantitative comparisons in fashion human ATR dataset and LSUN outdoor church dataset.

Metrics	ATR Dataset						LSUN Outdoor Church Dataset					
	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours	Ours (refine)	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours	Ours (refine)
FID ↓	75.72	69.7	79.02	77.44	54.47	**43.65**	34.61	39.34	40.15	39.85	30.70	**23.86**
SSIM ↑	0.79	0.80	0.82	0.80	**0.83**	**0.83**	0.79	0.79	0.80	0.78	**0.81**	**0.81**
PSNR ↑	19.34	20.55	21.98	19.84	22.84	**22.95**	21.22	18.93	21.26	18.21	19.99	**22.04**

TABLE II
Quantitative comparisons in Garment dataset and Cafidataset.

Metrics	Garment Dataset						CafiDataset					
	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours	Ours (refine)	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours	Ours (refine)
FID ↓	21.58	21.46	21.35	23.57	20.66	**6.21**	22.19	23.07	22.43	26.72	13.67	**6.73**
SSIM ↑	0.85	0.85	**0.87**	0.85	0.85	**0.87**	0.89	0.90	0.91	0.90	**0.92**	**0.92**
PSNR ↑	23.49	23.14	24.91	22.40	23.30	**25.15**	26.35	27.91	28.25	27.39	**30.53**	29.95

to handle multiple edits simultaneously. The first two rows display a challenging scenario in the ATR dataset involving irregular masks. In the first row, the left hand's pose is altered and the jeans are lengthened. The second row modifies the short pants into a short skirt while removing the watermark. In these complex editing tasks, our CoDE-GAN succeeds in generating the most plausible textures. On the other hand, E2I, SC-FEGAN, and DeFLOCNet exhibit artifacts on the masked background. Although Gated Conv can accurately represent the edited content, its synthesized clothing textures are less uniform than ours, especially when editing the geometric pattern on the floor in the second row. Fig. 7's third and fourth rows highlight CoDE-GAN's ability to handle simultaneous edits on two distinct areas (the sleeves) with discontinuous box masks, which lie outside the training distribution. Despite these challenging conditions, CoDE-GAN consistently synthesizes visually appealing edited images. On the other hand, E2I and DeFLOCNet tend to produce artifacts around the mask boundary. Both Gated Conv and SC-FEGAN struggle with the task, notably failing to correctly synthesize the cuff region in the third row and generating artifacts in the background.

2) Quantitative Comparison: Tab. I and Tab. II list the quantitative comparison with the state-of-the-art methods on four datasets. It is clear that our CoDE-GAN achieves competitive performance on all the datasets. In terms of the FID, our CoDE-GAN obtains remarkably better scores than other methods. This indicates that our method could synthesize the most realistic images. In addition to FID, our CoDE-GAN excels in maintaining structural and perceptual quality, as indicated by strong performance in the SSIM and PSNR metrics. Moreover, it should be noted that our proposed CoDE-GAN does not include the refined stage. For fair comparison with Gated Conv and E2I, we followed Gated Conv and used our proposed CoDE-GAN to train a refined model, which further improves the performance by a large margin. This demonstrates that our proposed CoDE-GAN can be easily combined with other methods to improve the generation performance.

C. Ablation Studies

TABLE III
Ablation results on the designed CDM and CEM modules.

CDM	CEM	FID ↓	SSIM ↑	PSNR ↑
-	-	69.63	0.8070	21.38
✓		56.87	0.8291	22.55
	✓	55.54	0.8276	22.68
✓	✓	**54.47**	**0.8331**	**22.84**

In this section, we perform ablation studies to analyze the effectiveness of each module of our proposed CoDE-GAN. To this end, we train a series of variant models on the ATR dataset: i) The Baseline model is the Gated Conv [10] without refine stage. ii) The +CDM is the Baseline model with encoder replaced by our proposed CDM. iii) The +CEM is the Baseline model which adds our proposed CEM. Tab. III and Fig. 8 respectively show the quantitative and qualitative results of the variant models and our full model. We can see from the results that our full model is superior to all the variant models. As Fig. 8 (c) shown, the generated image by Baseline reveals flaws when editing the dress length, producing incorrect textures. The CDM and CDM both perform better than Baseline in all metrics, but there are artifacts in the synthesized textures could be further improved. Particularly, compared with Baseline, Fig. 8 (d) shows that +CDM could help to improve the consistency between synthesized textures and unedited image content. On the other hand, +CEM could help to reduce less artifacts and allow the synthesized textures to be uniform. For example, the background artifacts in the second row of Fig. 8 (e) is reduced.in As we can see in Fig. 8 (f), our full model could generate reasonable edited textures.

Tab. III shows the quantitative results of ablation experiments carried out on the ATR dataset. The first row represents Gated Conv, our baseline model, on which our CoDE-GAN is built by applying CDM and CEM to its coarse network. The second and third rows show that our proposed CDM and CEM effectively reduce the FID score by about 25% compared to

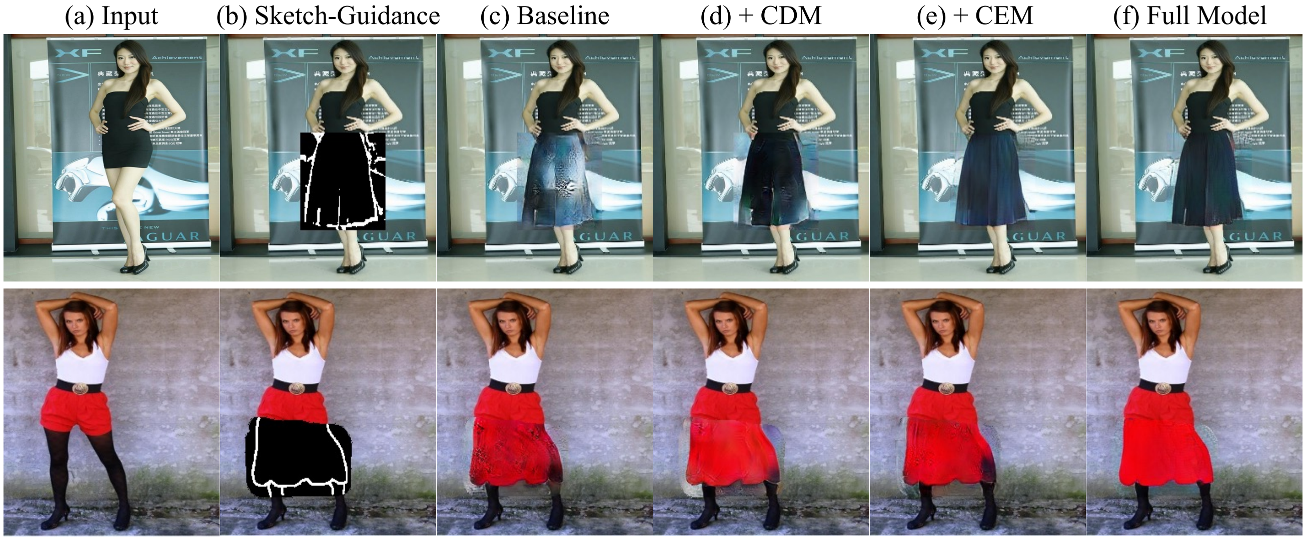


Fig. 8. Qualitative results for ablation studies.

the baseline. Notably, CDM primarily enhances SSIM, while CEM improves FID and PSNR. Combining these two modules together further enhances all of the metrics.

D. Discussions

In this section, we will discuss the impact on more possible settings in our proposed content-aware loss (Eq. 12), different preprocessing on the edge, and training masks.

TABLE IV
QUANTITATIVE RESULTS ON VARIOUS COMBINATIONS OF LOSS FUNCTION AND GROUND TRUTH TYPES.

Loss Function	Ground Truth	FID ↓	SSIM ↑	PSNR ↑
L1	Grey	17.88	0.87	25.24
Cos	Grey	**5.02**	**0.89**	**26.17**
L1	Segmentation	6.82	0.87	25.08
Cos	Segmentation	6.98	0.86	24.91
L1	Colour	17.57	0.87	25.13
Cos	Colour	15.77	0.88	25.29

a) Analysis on Content-Aware Loss: Tab. IV provides the comparison study of content-aware loss, evaluating various combinations of loss (Eq. 12) functions and ground truth types in the Garment dataset with 30% free-form masks. The optimal combination found involves using cosine similarity as a loss function to supervise the content response map with a grey image. This setup outperforms the rest by achieving the lowest FID score of 5.02 and the highest SSIM and PSNR scores of 0.89 and 26.17, respectively. When the L1 loss function is used with a grey image, we observe a decrease in performance. This is likely due to the lack of color information in the grey image, leading to variations in intensity across different images. On the other hand, cosine similarity helps normalize these intensity ranges, making it a more effective choice for content constraint. We also examined the effect of using foreground segmentation. Although this led to improvements in the metrics, it was not as generalizable as using a grey image, especially with complex datasets like the ATR dataset or LSUN outdoor church dataset. The binary segmentation is insufficient to represent different content regions in these cases, whereas the grey image can distinguish different contents through intensity variations. Finally, we considered the use of a color image for supervision but found the model challenging to optimize. The content response map, synthesized by the CEM from feature maps, is expected to resemble the final RGB output. This places a heavy burden on the CEM, potentially requiring a larger model with more parameters. This goes against our design motivation of maintaining a lightweight, efficient module to apply content constrain. Therefore, this combination did not yield optimal results.

b) Analysis on Edge Binarization: The sketch-controlled editing tasks require the user to input modified sketches. However, current sketch-controlled literatures typically utilize edges extracted via the Holistically-Nested Edge Detection (HED) technique as sketches, which often diverge from actual user inputs. To determine the influence of preprocessing methods on HED-extracted edges, we conducted a comparison study to assess the potential benefits of binarizing these edges to better mimic user-drawn sketches. The results of this investigation, presented in Tab. V, are based on the Garment Dataset. Each result was evaluated on the corresponding edge preprocessing. By binarizing the extracted grayscale edges with a threshold of 0.6, we observe a substantial deterioration in the quantitative metrics. In particular, the FID metric demonstrates a more than twofold increase for most methods, except for E2I. This suggests that E2I is robust to edge form variations, likely due to its two-stage network implementation, which incorporates edges at each stage. This structure emphasizes the importance of edges, aiding in the reconstruction of spatial structure. Our method, however, still achieves the best FID score of 5.02 on grayscale edges and 20.66 on binarized edges. The decrease in performance is attributed to information loss during edge binarization. The HED-extracted edges represent the likelihood that a given pixel could be an edge. Consequently, aside from actual edge pixels, there is a

TABLE V
QUANTITATIVE RESULTS ON EDGE PREPROCESSING METHODS.

Metrics	Binarized Edge					Greyscale Edge				
	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours	DeFLOC-Net [14]	SC-FE GAN [12]	Gated Conv [10]	E2I [11]	Ours
FID ↓	21.58	21.46	21.35	23.57	**20.66**	9.03	7.77	6.89	17.07	**5.02**
SSIM ↑	0.85	0.85	**0.87**	0.85	0.85	0.86	0.86	0.87	0.87	**0.89**
PSNR ↑	23.49	23.14	**24.91**	22.40	23.30	24.32	24.11	24.69	23.97	**26.17**

TABLE VI
EVALUATION RESULTS WHEN TRAINED ON FREE-FORM MASK WITH DIFFERENT RATIOS.

Mask Ratio	30%			50%			70%		
Metrics	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑
DeFLOCNet [14]	23.50 (5.84)	0.76 (0.90)	20.00 (27.27)	**14.10** (7.58)	0.81 (0.85)	22.14 (25.56)	13.39 (11.47)	0.81 (0.80)	22.52 (23.60)
SC-FEGAN [12]	24.92 (5.51)	0.78 (0.92)	20.91 (28.24)	21.72 (9.48)	0.79 (0.86)	21.44 (25.39)	19.29 (13.83)	0.80 (0.79)	22.07 (23.14)
Gated Conv [10]	25.92 (3.76)	**0.80** (0.93)	**21.88** (29.28)	29.25 (14.34)	**0.82** (0.88)	21.90 (26.69)	27.38 (20.81)	0.82 (0.82)	**23.11** (24.60)
E2I [11]	96.64 (15.84)	0.67 (0.92)	17.45 (28.07)	63.92 (32.26)	0.72 (0.79)	17.08 (21.38)	84.53 (93.62)	0.70 (0.60)	16.14 (16.05)
Ours	**22.55** (**2.79**)	0.78 (**0.94**)	21.19 (**30.62**)	14.13 (**4.76**)	0.81 (**0.89**)	**22.49** (**27.99**)	**12.45** (**7.10**)	**0.83** (**0.84**)	22.98 (**25.73**)

TABLE VII
EVALUATION RESULTS WHEN TRAINED ON BOX MASK WITH DIFFERENT RATIOS.

Mask Ratio	30%			50%			70%		
Metrics	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑
DeFLOCNet [14]	34.35 (9.03)	0.74 (0.86)	20.30 (24.32)	25.52 (18.45)	0.77 (0.77)	21.39 (21.32)	26.59 (27.48)	0.76 (0.71)	20.28 (19.51)
SC-FEGAN [12]	**22.63** (7.77)	**0.77** (0.86)	**21.97** (24.11)	**22.6** (14.14)	0.77 (0.77)	21.99 (21.13)	28.12 (21.11)	0.76 (0.69)	21.01 (18.95)
Gated Conv [10]	23.51 (6.89)	**0.77** (0.87)	21.04 (24.69)	23.77 (12.09)	0.78 (0.79)	21.74 (21.67)	**23.01** (15.90)	0.78 (0.72)	**22.02** (21.11)
E2I [11]	124.49 (17.07)	0.62 (0.87)	16.14 (23.97)	91.16 (33.91)	0.67 (0.75)	16.47 (18.48)	110.89 (78.77)	0.66 (0.59)	16.02 (13.70)
Ours	29.26 (**5.02**)	0.76 (**0.89**)	20.84 (**26.17**)	25.47 (**8.36**)	**0.79** (**0.82**)	**22.00** (**23.63**)	27.47 (**11.55**)	**0.79** (**0.75**)	21.52 (**21.91**)

higher probability assigned to pixels near the edge, providing crucial prior information about an image's spatial structure. Upon binarization, the data becomes too sparse to effectively guide the model in reconstructing spatial structure. Although grayscale edges allow models to perform well quantitatively, their characteristics do not resemble the naturalistic qualities of human-drawn sketches. Therefore, we adopt binarized edge training on all of the qualitative results.

TABLE VIII
RELATIONSHIPS TO MASK RATIOS AND REGION OF INTERESTS.

Ratio Type	Free-Form			Box		
mask-to-image	30%	50%	70%	30%	50%	70%
mask-to-foreground	34%	55%	75%	46%	74%	91%

c) Analysis on Training Mask Settings: The flexible editing may involve various mask types. To evaluate the impact on the robustness of different masks when trained on a specific mask type, we conduct comparison experiments on Garment Dataset. Tab. VI and Tab. VII presents results obtained from training with both free-form and box masks at varying mask ratios. Each result corresponds to a specific mask setting and is evaluated under six different mask configurations - two mask types (box and free-form), and three mask ratios (30%, 50%, 70%). Additionally, results evaluated with mask configurations that align with the training settings are also included in the brackets.

Overall, when the evaluation mask aligns the training settings, our methods achieve the best performance (see from the brackets results). When evaluated in all kinds of masks, we find it is beneficial to increase training mask ratio as it brings improvement to most of the methods. This is because that most of the methods only calculate loss on reconstructed (masked) region. A greater mask area would lead to much gradient information. However, we find the gain is limited when trained on box mask with 70% area (performance decreased in the last three columns of Tab. VII). Since the box masks are continuous, it is much likely to present on the foreground of the image. Therefore, less information is provided when reconstructing the masked region. Tab. VIII shows the corresponding masked foreground region. 70% box mask would cover more than 90% foreground region. In conclusion, it is optimal to trained mask ratio with about 70% in the foreground which is free-form mask with 70% ratio or box mask with 50% ratio. Comparing with other SOTA methods, ours CoDE-GAN is robust when trained on free-form mask. Its FID achieves the best in 30% and 70% mask ratio. The difference when trained on 50% is less 1% to DeFLOCNet.

V. FAILURE CASE ANALYSIS

Fig. 9 provides two examples where CoDE-GAN encounters difficulties. CoDE-GAN's primary objective is to edit the shape of clothing content with textures consistent with the unmasked (reference) region. The model learns to automat-

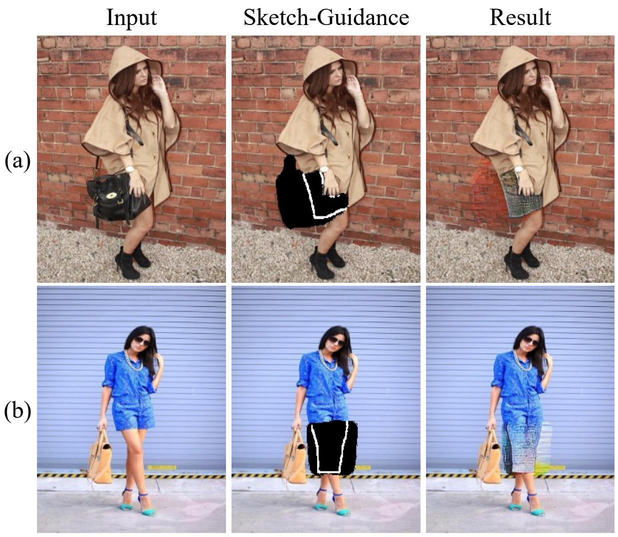
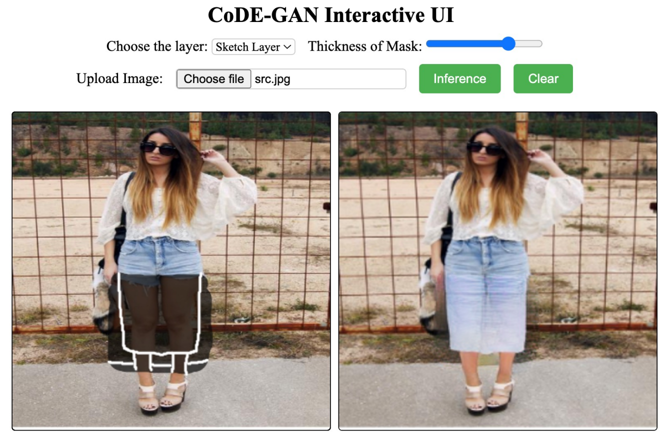


Fig. 9. Failure case in handling distracting background.

ically find the correct reference content through adversarial training. However, when the background content is distracting or similar to the foreground, our model struggles to perform satisfactorily. The first three columns in the example demonstrate the completion of the obscured coat and the removal of the bag against a brick background. While our CoDE-GAN method performs better on reproducing the background brick texture, it unfortunately falls short in synthesizing convincing clothing textures. The synthesized textures in the regions of interest are overly influenced by the background, causing an undesired shift towards background textures instead of the intended clothing. The final three columns attempts to modify short pants into an A-line dress. The similarity in color between the clothing and the background presents a challenge for the model, resulting in a failure to identify the correct reference color, despite the provision of sketches to outline the shape of the dress. Consequently, the synthesized dress textures align more closely with those of the ground floor. Therefore, while CoDE-GAN generally exhibits effective performance across numerous samples, it continues to face challenges when handling images with visually intricate or ambiguous backgrounds.

VI. Interactive Web User Interface

Fig. 10 presents an interactive web user interface (UI), specifically designed to enable users to easily edit images. Users can upload their own images onto the web UI, and using the mask layer, they can draw a transparent black mask to mark the area they wish to edit. Following this, users can sketch their desired edits on top of the image using the sketch layer. Once the desired edits have been input, users can click the inference button, triggering our backend CoDE-GAN model to generate the edited results. This user-friendly web UI leverages Python as backend service, with basic HTML and JavaScript forming the front-end interface. Notably, our Web UI has the potential to be a valuable resource for other sketch-controlled methods, providing the community with a tool that's both accessible and easy to use. The UI's web-based nature reduces the need for user-side installations, allowing access via a web browser. Unlike existing model (SC-FEGAN) that packages the UI and the model together into an executable file exclusive for Windows systems, our web UI can be effortlessly adapted to MacOS without any need for code modification or compilation. Furthermore, the model deployed as a backend service provides flexibility for other developers. They can use the API in similar applications, test different models with minor adjustments, and choose whether to deploy the backend services locally or globally, given the availability of a public IP address. The hardware requirement is a minimum of 4GB runtime memory, providing further flexibility for deployment.


CoDE-GAN Interactive UI

Choose the layer: Sketch Layer

Upload Image: src.jpg

Fig. 10. Interactive web user interface for allowing sketch-controlled editing.

VII. Conclusion and Future Work

In conclusion, we have proposed a new method, CoDE-GAN, to allow for flexible sketch-controlled editing of fashion image content with consistent textures. Our approach decouples content and texture representation to overcome the obstacle of reconstructing content regions contoured by sketches due to the lack of information within them. We have shown through experiments on the fashion human ATR dataset and garment-based Garment and CafiGarment datasets that CoDE-GAN achieves superior results compared to state-of-the-art methods in terms of perceptual quality and editing flexibility. CoDE-GAN has the potential to greatly improve the efficiency of image editing in the fashion industry.

As in future work, there are several directions that can be explored to improve the CoDE-GAN method proposed in this study. One possibility is to integrate the use of additional guidance, such as texture patches, to edit the clothing textures. Another direction is to incorporate more advanced generative models, such as Generative Flow models or Denoising Diffusion models, to improve the quality of the generated images. Furthermore, it would be interesting to explore the use of CoDE-GAN for other applications beyond the fashion industry, such as image inpainting or guided image reconstruction. Finally, it would be valuable to further evaluate the performance of CoDE-GAN on a wider range of datasets beyond the fashion dataset and the church dataset to demonstrate its generalizability.

REFERENCES

[1] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, "Language-based image editing with recurrent attentive models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8721–8729.

[2] Y. Li, Z. Gan, Y. Shen, J. Liu, Y. Cheng, Y. Wu, L. Carin, D. Carlson, and J. Gao, "Storygan: A sequential conditional gan for story visualization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6329–6338.

[3] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7880–7889.

[4] A. Frühstück, K. K. Singh, E. Shechtman, N. J. Mitra, P. Wonka, and J. Lu, "Insetgan for full-body image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7723–7732.

[5] Y. Li, Y. Li, J. Lu, E. Shechtman, Y. J. Lee, and K. K. Singh, "Collaging class-specific gans for semantic image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 418–14 427.

[6] S. Jiang, J. Li, and Y. Fu, "Deep learning for fashion style generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4538–4550, 2021.

[7] P. Zhang, L. Yang, X. Xie, and J. Lai, "Lightweight texture correlation network for pose guided person image generation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[8] F. Ma, G. Xia, and Q. Liu, "Spatial consistency constrained gan for human motion transfer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 730–742, 2021.

[9] X. Yuan, D. Tang, Y. Liu, Q. Ling, and L. Fang, "Magic glasses: from 2d to 3d," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 843–854, 2016.

[10] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4471–4480.

[11] S. Xu, D. Liu, and Z. Xiong, "E2i: Generative inpainting from edge to image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1308–1322, 2020.

[12] Y. Jo and J. Park, "Sc-fegan: Face editing generative adversarial network with user's sketch and color," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1745–1753.

[13] S. Yang, Z. Wang, J. Liu, and Z. Guo, "Deep plastic surgery: Robust and controllable image editing with human-drawn sketches," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 2020, pp. 601–617.

[14] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, J. Liao, B. Jiang, and W. Liu, "Deflocnet: Deep image editing via flexible low-level controls," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 765–10 774.

[15] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.

[16] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[17] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1415–1428, 2021.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[19] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[21] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, "Deep human parsing with active template regression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 12, pp. 2402–2414, 2015.

[22] L. Chen, J. Tian, G. Li, C.-H. Wu, E.-K. King, K.-T. Chen, S.-H. Hsieh, and C. Xu, "Tailorgan: making user-defined fashion designs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3241–3250.

[23] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[24] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "Viton: An image-based virtual try-on network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7543–7552.

[25] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[26] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward characteristic-preserving image-based virtual try-on network," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 589–604.

[27] H. J. Lee, R. Lee, M. Kang, M. Cho, and G. Park, "La-viton: A network for looking-attractive virtual try-on," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[28] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[29] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo, "Towards photo-realistic virtual try-on by adaptively generating-preserving image content," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7850–7859.

[30] Y. Ren, G. Li, S. Liu, and T. H. Li, "Deep spatial transformation for pose-guided person image generation and animation," *IEEE Transactions on Image Processing*, vol. 29, pp. 8622–8635, 2020.

[31] X. Han, X. Hu, W. Huang, and M. R. Scott, "Clothflow: A flow-based model for clothed person generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 10 471–10 480.

[32] A. Cui, D. McKee, and S. Lazebnik, "Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 638–14 647.

[33] S. Choi, S. Park, M. Lee, and J. Choo, "Viton-hd: High-resolution virtual try-on via misalignment-aware normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 131–14 140.

[34] P. Dhariwal and A. Q. Nichol, "Diffusion models beat GANs on image synthesis," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=AAWuCvzaVt

[35] L. Zhu, D. Yang, T. Zhu, F. Reda, W. Chan, C. Saharia, M. Norouzi, and I. Kemelmacher-Shlizerman, "Tryondiffusion: A tale of two unets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4606–4615.

[36] D. Morelli, A. Baldrati, G. Cartella, M. Cornia, M. Bertini, and R. Cucchiara, "Ladi-vton: Latent diffusion textual-inversion enhanced virtual try-on," *arXiv preprint arXiv:2305.13501*, 2023.

[37] Q. Dai, S. Yang, W. Wang, W. Xiang, and J. Liu, "Edit like a designer: Modeling design workflows for unaligned fashion editing," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3492–3500.

[38] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[39] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[40] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[41] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[42] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.

Zhengwentai Sun received his Bachelor's degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2021. He is currently a Master of Philosophy Student at The Hong Kong Polytechnic University, Hong Kong. His research interests include image generation, editing, and their applications in the fashion domain.

Yanghong ZHOU received the bachelor's degree from Fujian Normal University, in 2011, and the master's degree from the University of Electronic Science and Technology of China, in 2014. She completed a Ph.D. degree focusing on the research of fashion image understanding in 2019 at The Hong Kong Polytechnic University in Hong Kong. After graduation, she is currently a post-doctoral fellow at the same university. Dr Zhou's current research interests include deep learning, image understanding and image segmentation.

P.Y. Mok received the B.Eng. degree (Hons.) majoring in industrial and manufacturing systems engineering and the Ph.D. degree from the University of Hong Kong, in 1998 and 2002, respectively. She is currently an Associate Professor with The Hong Kong Polytechnic University. Her current research interests include fashion pattern engineering, fashion 2D and 3D CAD, digital human modeling, 3D scanning and sizing, cloth simulation, deep learning, computer generated textile, sketch and pattern designs, computer vision and computer graphics in fashion applications, advanced data analysis, and artificial intelligent applications in the fashion industry.